Ritme+ a tangible task alarm

Robin Venhuizen - s2114186

Graduation project

University of Twente

Creative Technology

Client: Alderick van Klaveren, Ritme

Supervisor: Jelle van Dijk

Critical observer: Armağan Karahanoğlu

02-07-2021

Abstract

Neurodiverse people often struggle with doing their daily tasks, as they may lack executive functioning and cognitive flexibility. Ritme is an app developed to help with this by working as an alarm. It forces the user to get up and do their task. To turn this alarm off, the user needs to scan a QR code located at the task location. The goal of this research was to create a prototype of a product for Ritme that decreases the amount of phone usage needed, while still retaining all functionalities of the original Ritme app. Tangible interaction was implemented as a way to improve user experience. During this project, the creative technology design process was used. Consisting of diverging and converging phases. Within this process, a similar brainstorming technique to object brainstorming was used to generate ideas which were later evaluated and iterated upon. This resulted in a product that connects to the Ritme app and takes over the function of sounding the alarm and scanning the location-specific part (originally the QR code). This product did not include all tangible interaction guidelines found to be important, however, most of these were not applicable in this case. The guidelines that were applicable and missing can be easily implemented in the prototype in future iterations. Overall, the client was satisfied with the final product.

Acknowledgement

First and foremost I would like to thank Jelle van Dijk, Armağan Karahanoğlu, and Alderick van Klaveren. Both for providing me with the opportunity to do this project and for providing me with helpful feedback and constructive criticism along the way.

Furthermore, I would like to thank Reynaldo Cobos Mendez, Edwin Dertien and the AssortiMens foundation. Reynaldo for assisting me in the creation of the UML diagrams, and Edwin and the AssortiMens foundation for helping me find suitable candidates for one of my user tests.

I would also like to thank all participants of the user tests for participating. The selfless act of being willing to help without expecting anything in return is truly appreciated.

Finally, I would like to express my gratitude to one and all, who directly or indirectly, have helped me get to where I am today. Thank you.

Table of contents

List of figures

1. Introduction

1.1 Project context

This research project is a collaboration between the University of Twente and Ritme. Ritme is a startup company created by Alderick van Klaveren to help people keep organized, especially those with mental health problems. The company was founded in 2016 and since it is a start-up, developments for future products are in constant development. Ritme's vision is to provide neurodiverse people with tools to help them in daily life and experience life to its fullest. Neurodiversity was first used by Blume [1], who used it as a general term for having a different kind of brain wiring. Ritme's main focus is on neurodiverse people, however, products can be sold to and used by everyone. Neurodiverse people, and especially people with ASD and ADHD, are used as a way to evaluate products, under the motto: "if they can use it, everyone can".

Ritme has previously worked together with the University of Twente. This was done as part of the "Design for Specific Users" project as part of the Industrial design programme. The final design that came out of the project is called the RitStick. More on this will be discussed in the state of the art section.

1.2 Definition of the current Ritme system

The Ritme app is the first of these tools created. The app activates an alarm at user-specified times at which the user needs to do a certain task. Unlike normal alarms, the Ritme app requires the user to go to the task location to turn the alarm off. This is done by requiring the user to scan QR codes that are placed at the task location when setting up the app. For example, the user might have a QR code in the kitchen that needs to be scanned when the alarm for doing the dishes gets activated. Figure 1 and figure 2 show how the system currently works using a UML use case diagram and a UML sequence diagram.

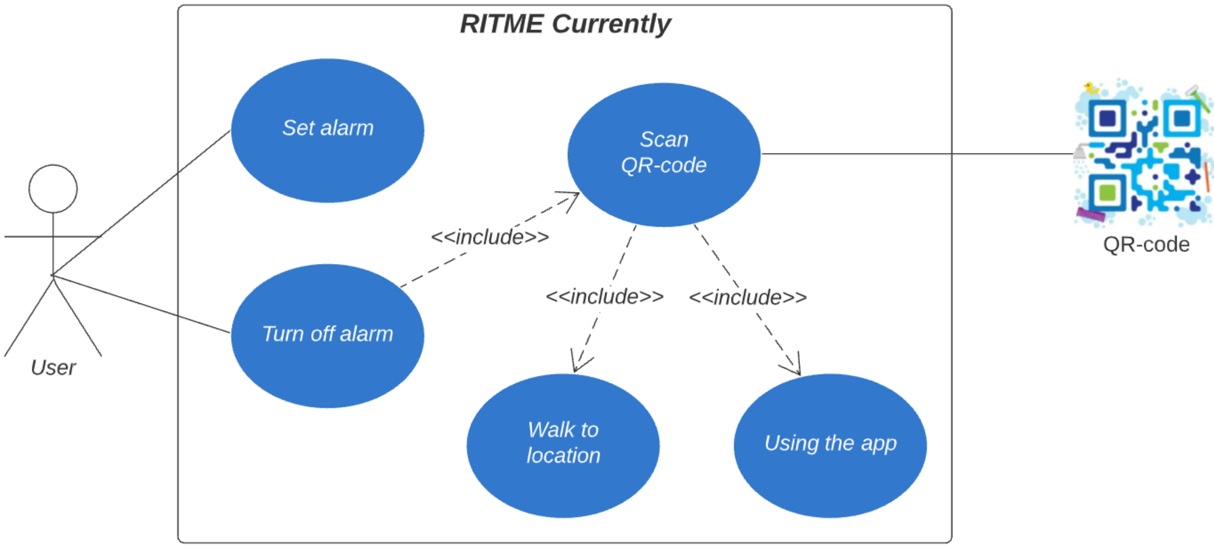


Figure 1 - Interaction between components of the current system and user shown in a use case diagram

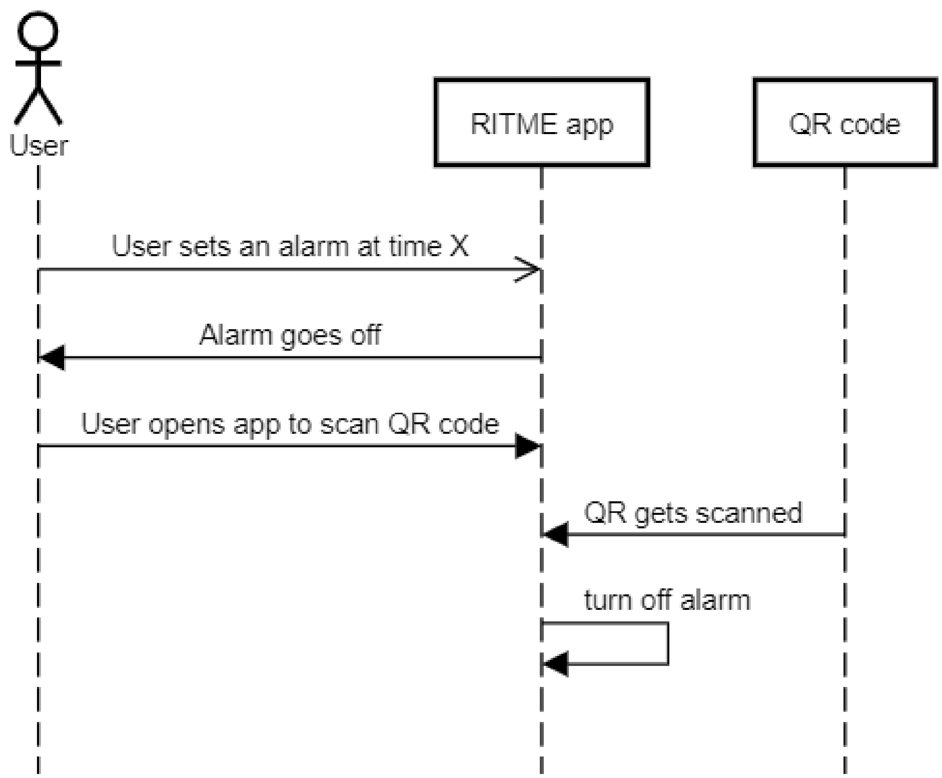


Figure 2 - Sequence diagram of the current system

1.3 Behaviour change techniques behind Ritme

One could say that in this way, the Ritme combines two of the behaviour change techniques described by Michie et al. [2], namely, Prompts/cues, and Non-specific incentives. The use of prompts/cues can be described as “using environmental or social cues as a stimulus for a certain behaviour”. For the Ritme, these cues are the alarms that turn on when a task is planned. Non-specific incentives are described by Michie et al. as “Inform that a reward will be delivered if and only if there has been effort and/or progress in performing the behavior”. While in this case no real reward is being offered, the user is rewarded for going to the task location indirectly by being able to turn the alarm off. According to self-determination theory (SDT) by Deci and Ryan [3], motivation by rewarding has a high impact on behaviour, however, it often reduces the amount of intrinsic motivation. Instead of seeing the interaction with the Ritme as a reward, it can also be seen as a way to avoid punishment, the punishment being the alarm. Based on SDT, Luria et al. [4] found that even though motivation based on avoiding punishment induces less intrinsic motivation, it still supports feature-focused, item-based memories that may support decisive action. In both cases (seeing the alarm as a punishment, or seeing it as an opportunity for reward), intrinsic motivation is not necessarily increased, however, user behaviour is impacted and decisive action is supported.

The behaviour of changing mindset and environment to do a task is related to set-switching and cognitive switching. Set-switching, first used by Jersild [5], is the ability to unconsciously shift attention from one task to another. Cognitive switching is very similar, the difference being that cognitive switching involves consciously shifting focus. The umbrella term for this is called cognitive flexibility, which is in itself a part of executive functioning. While set-switching isn’t necessarily a problem every neurodiverse person struggles with, for example, ADHD is not associated with difficulty in task switching [6], some neurodiverse people do find it difficult, for example, people with Autism Spectrum Disorder [7]. By requiring the user to move to the task location Ritme helps people with cognitive flexibility, especially cognitive switching. A normal alarm is easy to turn off and ignore, but requiring the user to go to the task location sets in motion the task completion process, therefore it is harder to return to behaviour not related to the task.

This is a similar principle as used by van Dijk et al. [8]. They developed a system in which controllable Philips hue lights can be programmed to light up at certain times. In this way, the lights function as a subtle reminder to start a task. Van Dijk et al. state “We saw the lights as distributed attention grabbers in the environment”. For the Ritme, the QR codes and the alarm combined function as the attention grabbers. The difference between the two is in the way the user is engaged to start. We can look at the matrix developed by Tromp et al.[9] for this, seen in figure 3. In the case of the system developed by van Dijk et al. [8], the design is persuasive, the user is not forced to do the task, they can easily ignore it if needed. In the case of Ritme, the design is Coercive. The alarm will not stop unless the correct QR code is scanned.

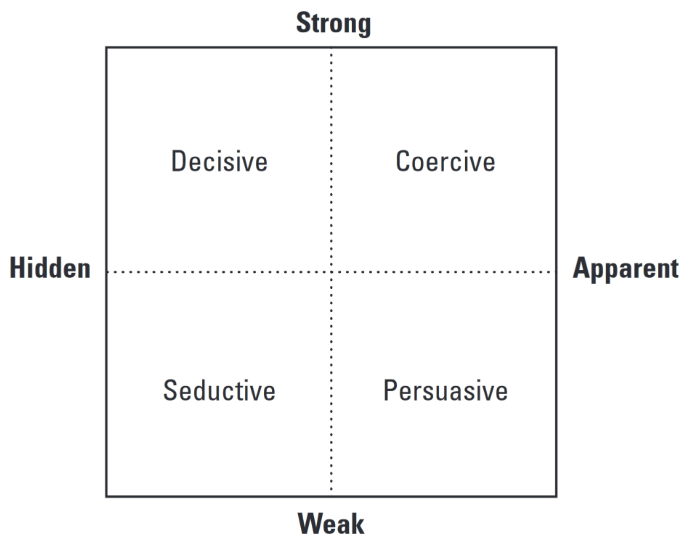


Figure 3 - Four types of influence based on the dimensions of force and salience [9]

1.4 Design Challenge

The main problem that is still present in the Ritme app is the fact that smartphones are still required. Smartphones are not ideal for several reasons. Firstly, smartphones are distracting which is not ideal for the user. When you want to get something useful done, you don't want to get distracted by the thing that is supposed to help you. Duke and Montag [10] state that phones not only negatively affect work productivity but also daily life outside of work. Dora et al.[11] performed several experiments to find determinants that influence switching from labour to leisure. Labour being actively involved in cognitive work and leisure being interacting with a smartphone. Two determinants that were found are the motivation for the given task and the presence of smartphone notifications. An increase in task motivation decreased the amount of leisure time while an increase in the number of notifications decreased labour time. Therefore, by decreasing the need for smartphone usage, the amount of labour time can be increased.

Secondly, by removing the necessity to use a smartphone, more emphasis is placed on the physical environment, therefore, increasing the amount of guidance. Using the physical environment to guide actions can be referred to as tangible interaction. For example, Djajadiningrat [12] describes this as "interaction with physical objects can exploit mankind's sophisticated perceptual-motor skills". He further notes that the human body is currently seen as a mechanical tool used by the superior mind. And that this type of reasoning results in an increasing emphasis on cognition and a decreasing emphasis on bodily movements. This results in usability issues as distinctions and meaning behind input for different outputs are hard to find. For this reason, it may be interesting to explore the use of tangible interaction and use its power to increase the usability of the Ritme.

Finally, an issue related to the development of the Ritme, Alderick mentioned that due to the variability in smartphones, software, operating systems etc., when a problem occurs it is hard to pin down where the problem originated. Therefore, by decreasing the influence of the smartphone these issues will be easier to resolve. For these three reasons, the main challenge is to reduce the phone usage needed to use Ritme through the use of principles found in tangible interaction research.

Before starting on designing the solution for this problem it is important to keep in mind that neurodiverse people, for example, people with autism, are likely to require different interaction methods than non-neurodiverse people. Therefore, a literature review will be conducted on the design requirements to consider when designing for neurodiverse people.

1.5 Proposed systems

One way to decrease the necessary smartphone usage to use Ritme is by creating a system that sits in between the smartphone and QR code. See figure 4. This system should take over some features that the smartphone is currently needed for, for example, scanning the QR codes. While it may be the case that QR codes will still be used. It is also possible these will be replaced by some other type of information containing item. Therefore in the rest of this paper, they will be referred to as “beacons”. This system will only include the functionally necessary features, apart from that it will not include any features, as to not be a distraction itself.

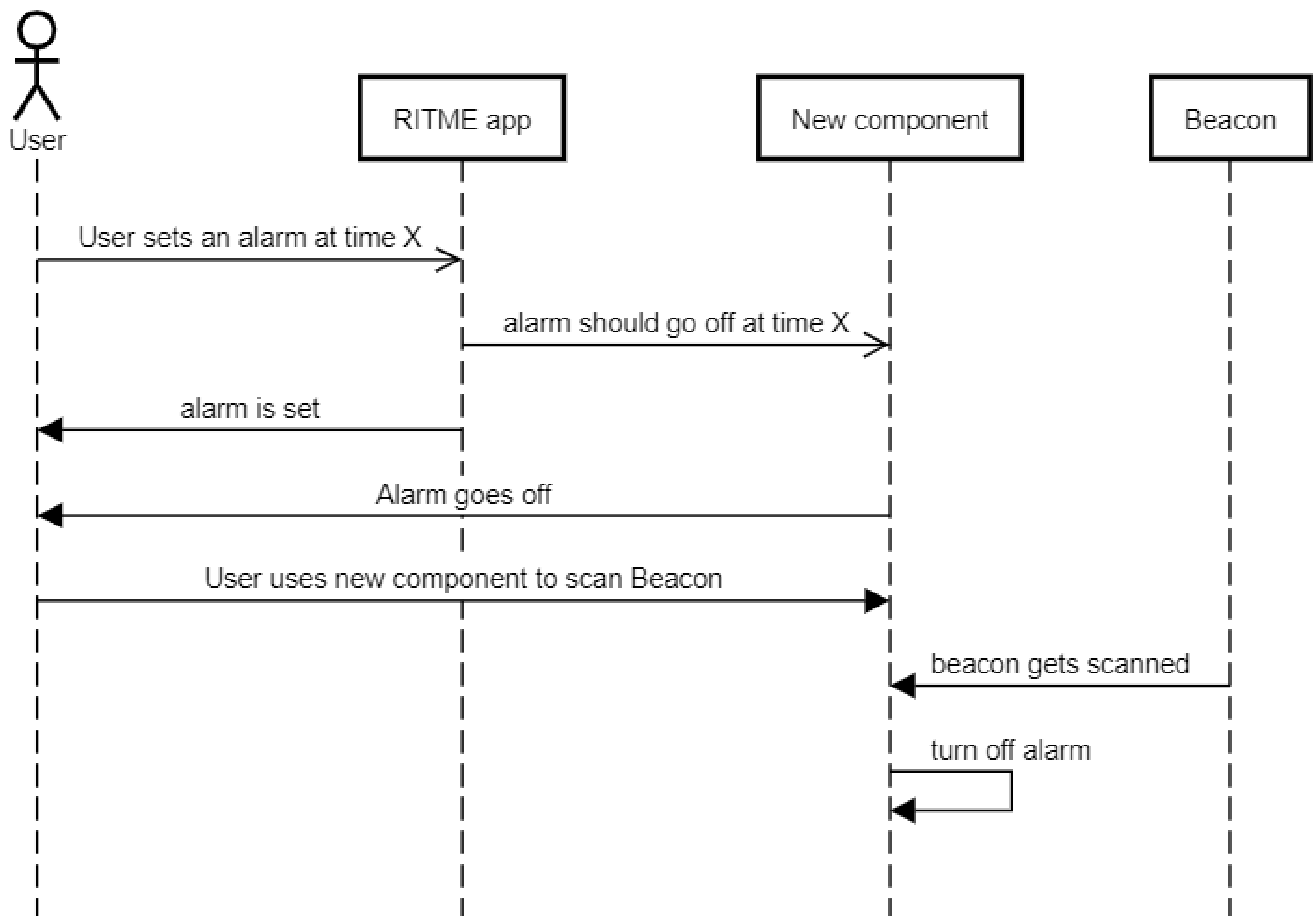


Figure 4 - The interaction between components for a system with an intermediary component

While it is clear that smartphone usage should be decreased, it is not clear in what way this needs to be done. For example, perhaps a device could be created that can replace all the smartphone’s features, making the smartphone arbitrary, see figure 5. Furthermore, it could be the case that an idea is created which only requires the use of beacons, for which the smartphone or intermediate device is not necessary, see figure 6.

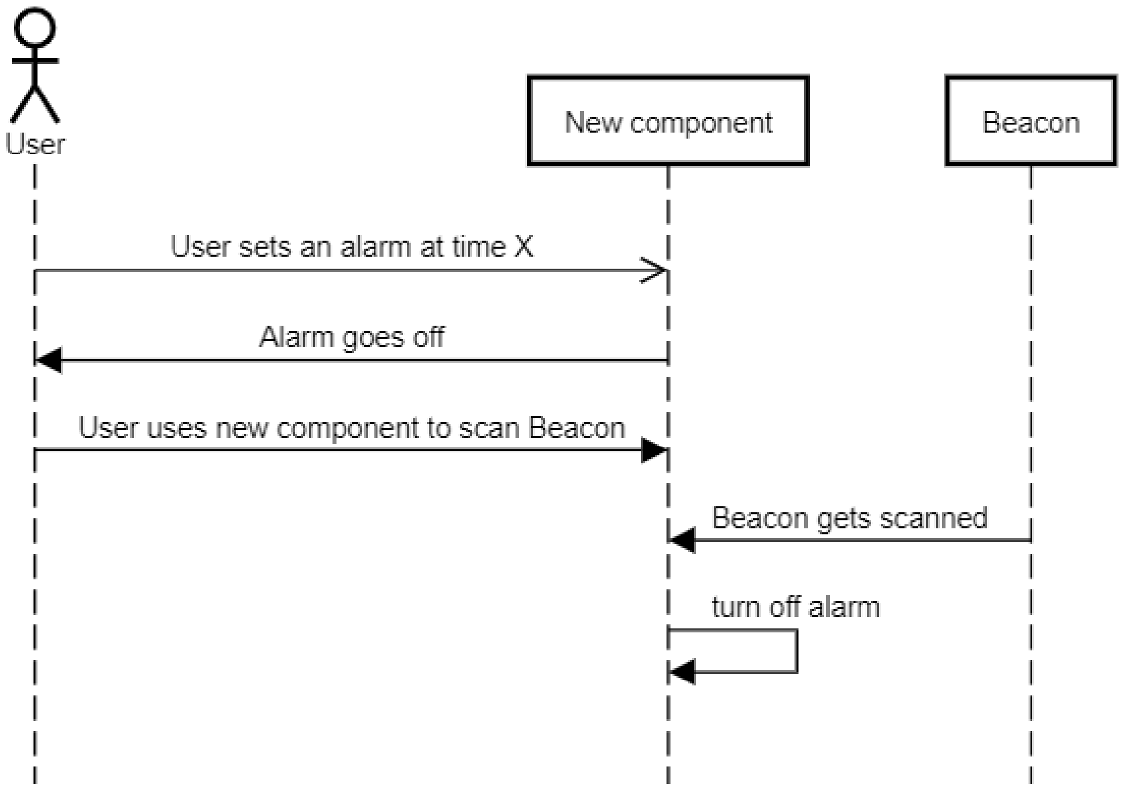


Figure 5 - The interaction between components of a system without the smartphone

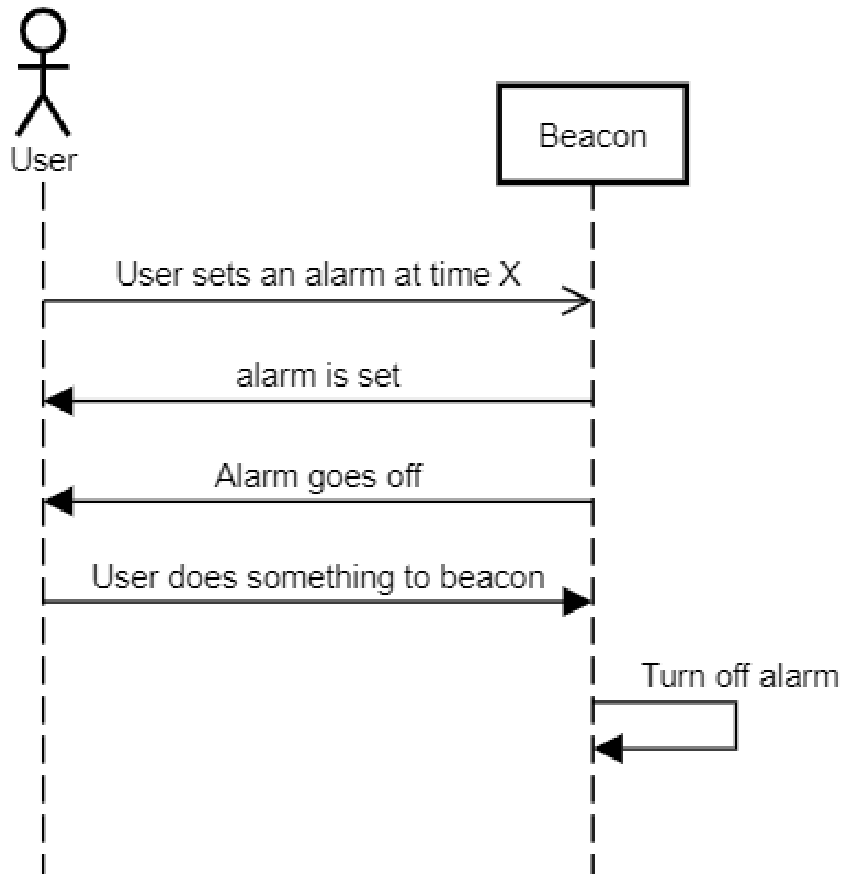


Figure 6 - The interaction between components of a system only consisting of beacons

These ideas will be evaluated in the ideation phase of this research and the best solution will be chosen. After finding the best solution in terms of user experience it may happen that it is not feasible to create the solution, at least in the given time frame. If that happens, a compromise will need to be made between user experience and technological complexity.

Finally, the form factor of the eventual solution should be considered. If the new component should be portable, how portable should it be? Should it be able to be carried in a pocket or a backpack? Should it be lightweight or does it not matter? This is important because, like with all design decisions, if the user doesn't use it, the product is useless. Therefore, these criteria will be investigated by performing user tests.

1.6 Research Questions

In summary, the challenge of this graduation project is to create a functional prototype by finding out how a product can be created that improves the user experience of Ritme by reducing smartphone usage and by making use of the power of tangible interaction design.
This challenge will be addressed by answering the following research questions:

1. What, if any, are important design guidelines or requirements to take into consideration when designing for people with neurodiversity?

2. In what way can the interaction design of Ritme be improved to be less dependent on smartphone usage?
   - 2.1. What are the practical constraints to make a realistic design proposal?
   - 2.2. How many of the app's features can be replaced by the physical product?
     - 2.2.1. How many of these features should be replaced
   - 2.3. What components does the eventual system consist of (only beacons/intermediary device and smartphone/only intermediary)

3. How can principles of tangible interaction be used in Ritme to improve user interaction?
   - 3.1. What are the main principles of tangible interaction?
   - 3.2. What form factor is best suited for the application, given the choice at sub-question 2.3?
   - 3.3. For each of the features in sub-question 2.1.1, what type of interaction is preferred, according to research and users?
   - 3.4. When and how should feedback be given to the user?

2. State of the Art

Before starting on the design cycle, it is good to know what has already been done in terms of devices, technique, technology and design process. Doing this may create ideas for this project, show research areas that require additional research, or show that research questions need to be improved.
This state of the art will consist of the following sections:

1. Design for neurodiversity
2. Tangible interaction
3. Planning tools (for neurodiversity)

Finally, requirements for the final product based on this literature research will be summed up.

2.1 Design for neurodiversity

As the term neurodiversity is very broad and this research is limited in time, this research cannot explore all design requirements for neurodiverse people. Ritme's focus is mainly on people with Autism spectrum disorder (ASD) and people with Attention deficit hyperactivity disorder (ADHD), therefore, this paper will only focus on these two groups.

2.1.1 Design for people with ASD

Part of the "Academic writing" course of Creative Technology is a literature review. It was highly encouraged to choose a topic related to your chosen graduation project, therefore I chose to do a literature review on product design for people with ASD. Therefore, the following section will mostly comprise information found in the academic writing course.

2.1.1.1 Software requirements

For the design of software, two factors are important to consider. The first and most important factor is customizability. People with autism often tend to monotropism [13], meaning that their interests are narrow. This results in them having less focus on things that are not related to said interests. Customizability, for example, the ability to change notification sound, allows users to adjust to their preferences, therefore making monotropism less of an issue as the software can be customized to fit the individual's interests, maintaining their focus. Customizability and personalisation are also mentioned as important requirements by [14]–[18].
Customizability is important for the stimuli the user experiences. Two important factors regarding stimuli were found. Firstly, stimuli should not occur randomly throughout the application. They should accompany consistent and predictable elements [16]. Furthermore, it is important to

consider whether stimuli are necessary. People with ASD, especially adults, do not like extra stimuli (e.g. animations), which do not serve any purpose other than looks[19]. Therefore, during the design of Ritme, it is important to consider the amount of customization possible and the stimuli used.

The display of information should also be evaluated. Fabri states, “information needs to be presented in a clear and objective fashion for it to be both informative and believable” [19]. Furthermore, according to Fabri, negative depictions of autism end up causing anxiousness and discouraged engagement. Finally, Fabri states, repeated content, typographic errors, and inappropriate language should be avoided at all costs as they severely undermine the perceived reliability of the information given. These three guidelines will also be taken into consideration during the design process of the Ritme wherever applicable.

2.1.1.2 Hardware requirements

Compared to software, guidelines for hardware are significantly less abundant. This is because most papers related to product design and ASD are about the process of creating a game of some sort, which only include software. Nonetheless, two requirements are described. Rasche [20] mentions that large buttons are preferred over small buttons, this is backed up by Putman et al. [14] who state that input devices should be easy to use, for example through the use of voice commands. Furthermore, they mention people with ASD prefer devices that are more portable over static devices. While these two hardware requirements are important, they do not relate well to the Ritme. Firstly, since the Ritme originally was a mobile application, to keep Ritme’s simplicity, this project does not intend to create something that is not portable, so that requirement is a given. Furthermore, tangible interaction will be explored within this project, therefore the second requirement of having large buttons is also redundant, as the entire goal of the project is to improve the user-product interaction, and buttons do not necessarily relate well to tangible interaction.

2.1.2 Design for people with ADHD

Literature on the guidelines for designing for people with ADHD we're much harder to find, as most research seems to focus on diagnosis and treatment, however, one literature review on the subject was found by McKnight [21]. This literature review used general guidelines for interacting with people with ADHD as a way to create guidelines for software design. It should be mentioned that these guidelines are based on children with ADHD and therefore may not apply perfectly to adults.

2.1.2.1 Software guidelines according to McKnight

Guidelines for designing for people with ADHD are found in a paper by McKnight [21]. These guidelines are listed in table 1. The guidelines that can be implemented within the design of the Ritme are highlighted in green and discussed further.

#	guideline
1	*The layout should be neat and uncluttered*
2	*Provide a 'calm' environment, with soothing colours. No decorations or distractions.*
3	*Provide a high reinforcement environment*
4	*Organise items in an orderly way*
5	*Distinguish important information by putting it in bold or colour. Signpost sections and group related information into panels.*
6	*Use a large font and a clear font.*
7	*Help pupils follow text by writing/highlighting alternate lines in different colours.*
8	*The usage of markers*
9	*Use brief and clear instructions*
10	*Allow ample rest periods and exercise breaks*
11	*Have a workstation that is enclosed, in a soundproof environment, with few distractions around*

12	*Keep technology away except during use*
13	*Keep to a routine*
14	*Minimise surprises*
15	*Maintain eye contact*

Table 1 - Software guidelines when designing for people with ADHD

In line with requirements found for designing for people with ASD is the fact that, when designing for people with ADHD, a well-organised layout should be used. Items on a page should be consistently placed and aligned to each other. Along with that, software should be calm and refrain from using surprises. To further improve user experience, a font should be chosen that is easy to read, McKnight [21] suggests using a sans-serif font like Arial. She further recommends a font size between 8 and 12. Although this should be adjusted to fit the screen size of the interface. Finally, McKnight [21] suggests that markers can be used to indicate to the user where in the process they are. These guidelines will be taken into account when designing the tangible product.

It is important to mention that all the guidelines given, relating to both ASD and ADHD, focus on the negative aspects of being neurodiverse. There seems to be a gap in knowledge relating to designing for neurodiversity taking into account the positive characteristics that neurodiverse people have, especially related to ADHD. For example, Putnam et al. [14] found that the strengths of people with ASD include: reading, math, memory, and a desire to be social. These characteristics may be used as a starting point of a design process. Similar knowledge is lacking in the case of designing for people with ADHD. It is therefore clear that a focus is placed on “designing for disability”, as a change of pace future research may shift that focus to “designing for the abilities of individuals with disabilities”.

2.2 Tangible interaction

As stated earlier, tangible interaction will be explored to increase the user experience for the Ritme. Tangible interaction is a way to describe the interaction we perform with physical products [22] or how we use the physical world around us for digital processes [23]. The problem with the digital processes used currently is that there is a big difference between the way we interact with our physical environment and the graphical user interfaces (GUIs) often used in digital tools. By limiting ourselves to the digital world, we can not use the flexibility of human movement to its full extent. Take for example a knob that can be turned to adjust the volume. It is much easier to accurately adjust the volume with a physical knob compared to a digital knob used for example in digital music production software. The Modsy is one of such tools that bridge the gap between digital and physical [24]. It uses physical knobs to control the digital environment of a digital audio workstation (see figure 7), creating an improved user experience.

Figure 7 - The Modsy controller [24]

While the Modsy does bridge the gap between physical and digital, tangible interaction is more than this. Tangible interaction is also about the shape of the product fitting in with the possible uses. In this way, a user knows what actions the product can perform, just by looking at its shape. An example of this is the video recorder by Overbeeke et al. [25] (figure 8).

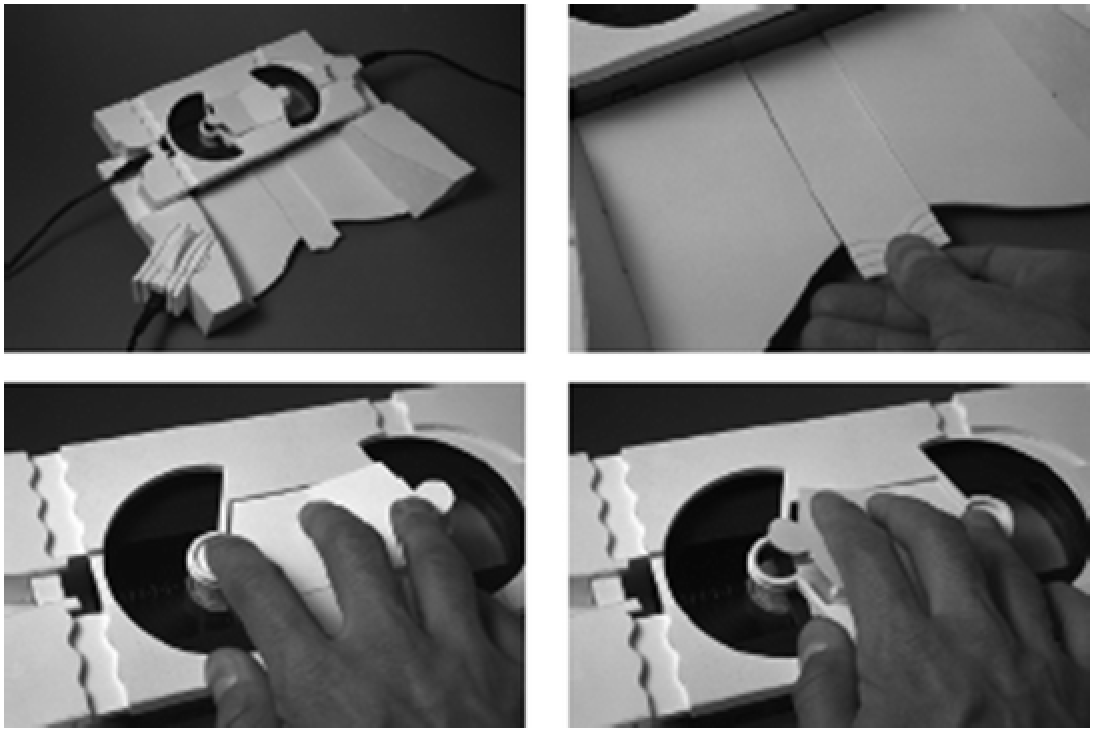


Figure 8 - The action of the user opens up the functionality. Starting top-left clockwise: the cassette remains visible whilst in the machine, pulling a ribbon triggers eject, and fast-forward/reverse becomes intuitively clear through a toggle placed between the tape reels. [25]

2.2.1 Tangible user interfaces

Tangible user interfaces or TUI is an alternative way to user-computer interaction. Ishii states: “Instead of making pixels melt into an assortment of different interfaces, TUI uses tangible physical forms that can fit seamlessly into a users' physical environment.” [23, p. 16] Tangible interface design uses the physical environment of the user as a representation of information and as a way to control the digital environment, however, Ishii continues to say that tangible interfaces do have limitations. In contrast to pixels used in GUIs, the physical atoms used in tangible user interfaces are not very malleable. It is harder to move and change a physical object compared to a digital object on a screen. Therefore, to complement the tangible user interface, intangible aspects, like sound and video, can be used, for example, by giving the user feedback on the action they just performed.

2.2.1.1 Principles of tangible user interfaces

Ishii [23] states several key properties for the design of tangible user interfaces. The most important part is the mapping of digital elements to physical objects and the other way around. The form of the controller used for this physical interaction is highly dependent on the intended use of the product, if a more abstract form is used, for example, disks, then giving feedback through intangible designs is more important, as it is more difficult to tell whether a certain action had an effect. Furthermore, the interaction with these controllers must be designed in such a way that the actions to be taken are based on well-understood actions related to the shape. "This understanding of the culturally common manipulation techniques helps disambiguate the users' interpretation of how to interact with the object." [23, p. 18] Furthermore, to complement the tangible objects, intangible features can be used to increase the user's perceptual coupling. With this, it is very important to have continuity between the tangible and intangible parts, as a lack of continuity will result in loss of immersion. Additionally, tangible user interfaces can take advantage of using multiple feedback loops. One feedback loop will occur through intangible features such as screens, while a second feedback loop will occur through tangible features which offer immediate tactile feedback. It is possible to further increase the number of feedback loops in a system by having the computer actuate the controllers. This is shown in figure 9.

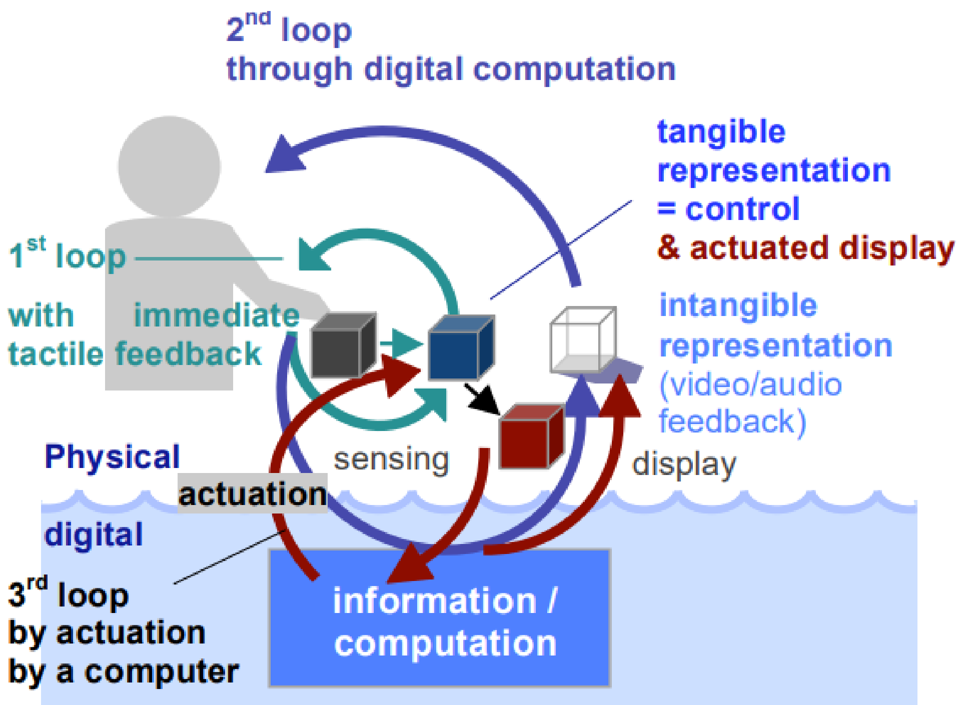


Figure 9 - The feedback loop possibilities of a TUI [23]

Another feature of TUI is the use of space-multiplexed input instead of time-multiplexed input. GUIs use time-multiplexed input, meaning that one device is used as a controller which sequentially can perform a multitude of tasks. In contrast, space-multiplexed input consists of multiple controllers that can be used simultaneously allowing collaboration between multiple people [23].

2.2.2 Framework of tangible interaction

The tangible interaction framework created by Hornecker and Buur [22] serves as a way to find requirements needed for successful tangible interaction (figure 10). The columns "spatial interaction" and "embodied facilitation" are both concerned with the social interaction aspect of tangible interaction. Since this does not concern Ritme, these will not be discussed in this research.

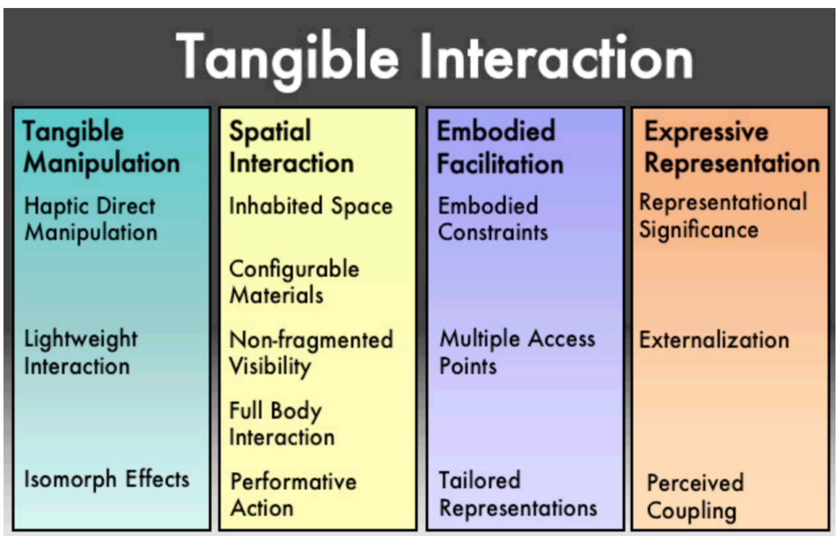


Figure 10 - Tangible interaction framework by [22]

2.2.2.1 Tangible manipulation

Tangible manipulation relates to the physical interaction with objects to change digital resources. These physical objects are directly related to the object of interest, unlike a mouse which acts as a general interaction method for multiple objects. Hornecker and Buur refer to this as "haptic direct manipulation" [22]. "One manipulates the interaction objects, has tactile contact, feels haptic feedback and material qualities." This method of interaction attracts the user by stimulating multiple senses. However, only having a good physical representation of the digital object is not enough, lightweight interaction is also needed in the form of constant feedback. This allows the user to proceed in small steps. Lastly, as also stated by Ishii [23], the connection between actions and effects should be clear (Isomorph effects). Therefore, the interaction with the physical object should closely resemble the actions performed digitally, in form, movement and time. One addition to this that Hornecker and Buur make is that it is not always necessary to have one-on-one mappings. Metaphors or other representations of data can be used for both input and output as a way to enliven interaction.

2.2.2.2 Expressive representation

Expressive representation relates to how a system can use physical things to explain its use, without the need for digital representations. In this way, the user can use physical aspects of the system as an aid in thinking, or as a means to recall certain actions taken before. Furthermore, Hornecker and Buur[22] state that if the goal is to create a tangible interface, the tangible parts of the system should be vital to the use of the system. If this is not the case, users will not see the system as tangible. “users perceive a tangible interface as “not very tangible” and the tangible objects as insignificant, if these were only of temporary relevance or not expressive” [22, p. 441] The feeling of the system being tangible are further enhanced by having a strong “perceived coupling”. A clear coupling should be present between what actions the user takes and what actions the system responds with.

2.2.3 Coupling of action and reaction

To ensure a clear coupling between the tangible and intangible aspects of the product, the “interaction frogger” framework by Wensveen et al. [26] can be used. In a mechanical product (no digital elements), coupling between action and reaction is a given, resulting in intuitive products. Wensveen et al. distinguish six characteristics of natural coupling. The action of the user and the reaction of the product should occur: at the same time, at the same location, in the same direction, they should be dynamically related (e.g. a smooth motion resulting in a smooth reaction), their sensory modalities should be in harmony, and the expression of the action should reflect the expression of the reaction.

These six types of unity of mechanical products can be translated into guidelines for intuitive technological designs. When designing interaction, one can strive to fulfil all these aspects whenever possible. However, as Wensveen et al. mention “as more functionality is added to electronic products full unification on all the aspects may be difficult or even undesirable to achieve because intuitive interaction needs to be balanced with technology, ergonomics, production costs or aesthetics.”[26, p. 179] Meaning that careful consideration should be done on whether increasing the intuitiveness decreases other aspects of the product and whether that trade-off is worth it. If a clear coupling between action and reaction is not possible, further information should be provided.

2.2.3.1 Ways of giving information

In the interaction frogger framework [26], Wensveen et al. distinguish between feedback and feedforward as ways to give information to the user. “Feedback is the information that occurs during or after the user’s action. But before the user’s action takes place the product already offers information, which is called feedforward” [26, p. 180]. Within this distinction, a further distinction is made between inherent, augmented and functional.

Inherent- feedback and feedforward are both related to the motor skills of a person. For feedback, an example is the feeling of pushing a button. You feel the resistance and texture of the button and the eventual click that follows. Focus is increasingly also put on these aspects of design instead of just appearance, therefore it is important to also consider this when designing a product [26]. Inherent feedforward shows the user how to use the product before using it, similar to expressive representation as described by [22].

Augmented feedback and augmented feedforward both come from another source than where action and reaction occur. For feedback, an example of this is an LED that turns on when a tv turns on. Since the TV does not turn on instantly, the designers of the TV added an LED to indicate the state of the system. Wenserveen et al. state “... this kind of feedback is usually added to inform the user about the internal state of the system … It can indicate ‘stand by’, ‘waiting’, ‘sleeping’, ‘processing’ etc.” [26, p. 180]. Augmented feedforward is information that is provided, from another source, for example, stickers or labels.

Finally, functional- feedback and feedforward. Functional feedback is the most basic kind of feedback. It is the feedback that is created by the system functioning. For example, you hit play on your stereo system and music starts playing. Functional feedforward is similar in the way that it is related to the function of the product. It shows the user the functionality of the product [26].

2.2.4 Examples of tangible interaction

In this section, several examples of tangible interaction will be given. The first of these is the "SandScape" created by MIT Media Lab's Tangible Media Group [27]. It is an interactive sand-filled tabletop. Users can change the landscape by physically interacting with the sand. A projection will then show their impact on height, shadows, drainage, and more in real-time. See figure 11.

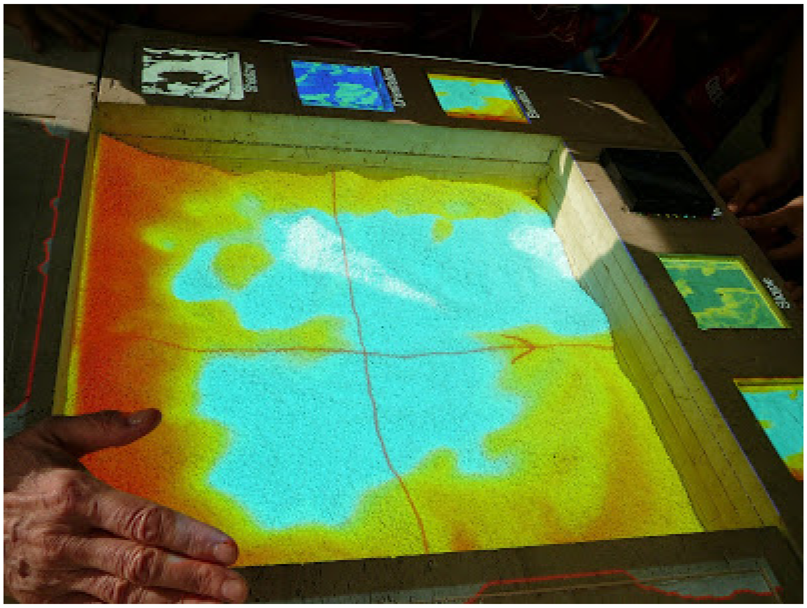


Figure 11 - Sandscape by Tangible Media Group [27]

A second example of tangible interaction being used to enhance user experience is the Microsoft Surface Dial [28], see Figure 12. The surface dial can be used on any Microsoft surface product. The surface dial can for example be used to scroll and adjust volume, however, when placed on a compatible screen the most impressive features become apparent. It can then for example be used as a colour picker in art programs

Figure 12 - Microsoft Surface Dial

Another example is “Siftables” developed by Kalanithi and Maes [29]. “Siftables consists in a collection of compact tiles (36mm x 36mm x 10mm) - each with a colour LCD screen, a 3-axis accelerometer, four IrDA infrared transceivers, an onboard rechargeable battery and an RF radio.” See figure 13. The Siftables can be used for a multitude of applications. An example they give is a photo sorting task. Where users sort Siftables based on the image they display. The siftables can detect whether they are sorted correctly and say when sorting is done.

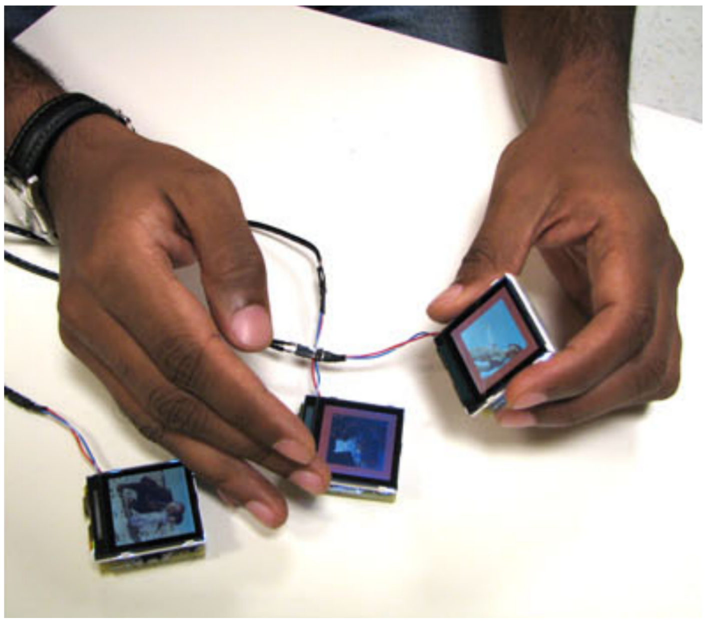


Figure 13 - Siftables [29]

2.3 Current planning tools (for neurodiversity)

In this section, several planning tools will be discussed. Starting with the Ritme, and the RitStick. After which an overview will be given of other planning tools that exist (either specifically targeted to neurodiverse people or not).

2.3.1 Ritme

As stated in the introduction, Ritme currently works with an app and QR codes. The user sets alarms and assigns these to the appropriate QR code. This section will discuss features that Ritme currently has, which should not be lost when going for a tangible interaction approach.

One of Ritme's features is that users can choose between setting a "reminder" and a "quick reminder", the difference being that normal reminders work based on a set time, for example, 11:48 AM, while quick reminders work based on a countdown. A comparison between these two can be seen in figure 14. A second feature of Ritme is that users can set locations for where reminders should occur. For example, a user might only want reminders about dishes when he is at home. The Ritme uses the GPS of the user's phone to check whether they are at the location and reacts accordingly. Because GPS is limited in accuracy, a very specific location like "in the bathroom" cannot be used, instead, more general locations like "work" and "home" are used. Thirdly, the Ritme does not require their proprietary QR-codes, it works using any QR code, however, the official Ritme QR codes are waterproof, which makes them preferred over other QR codes.

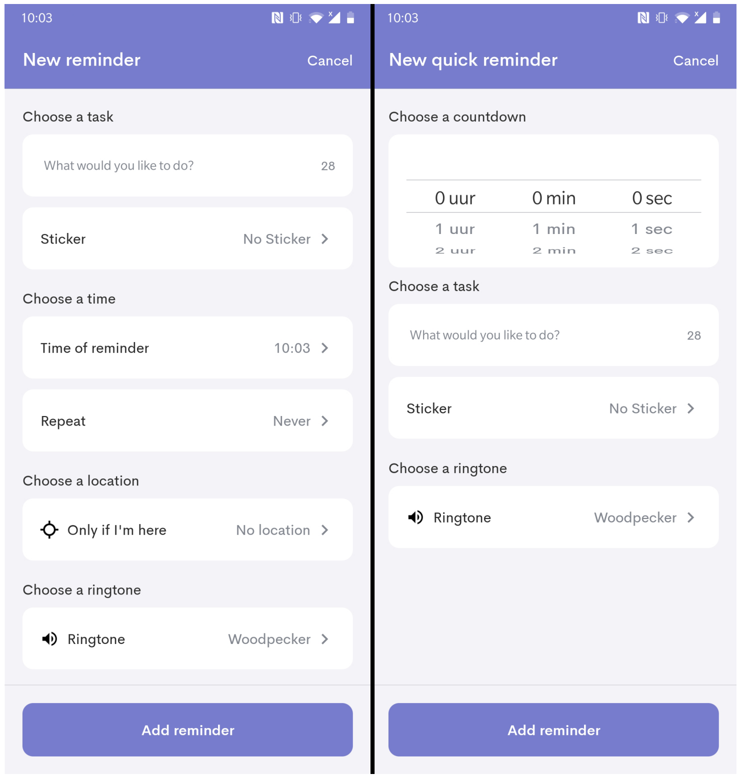


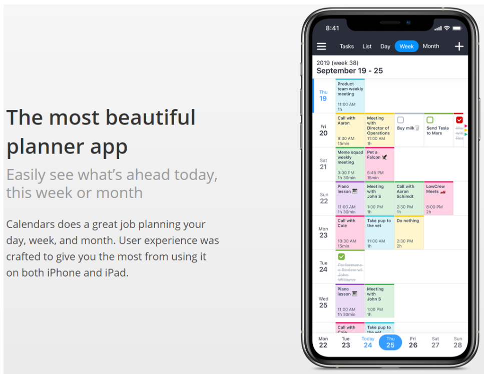
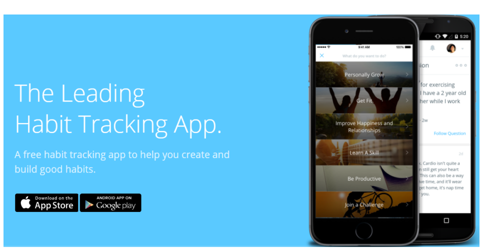
Figure 14 - Normal reminder compared to a quick reminder in the Ritme app

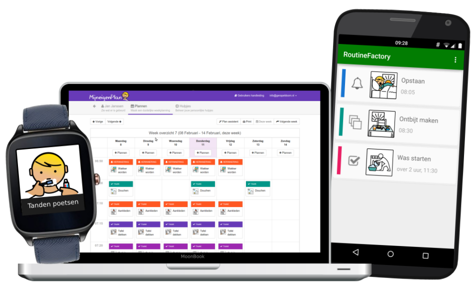
2.3.2 RitStick

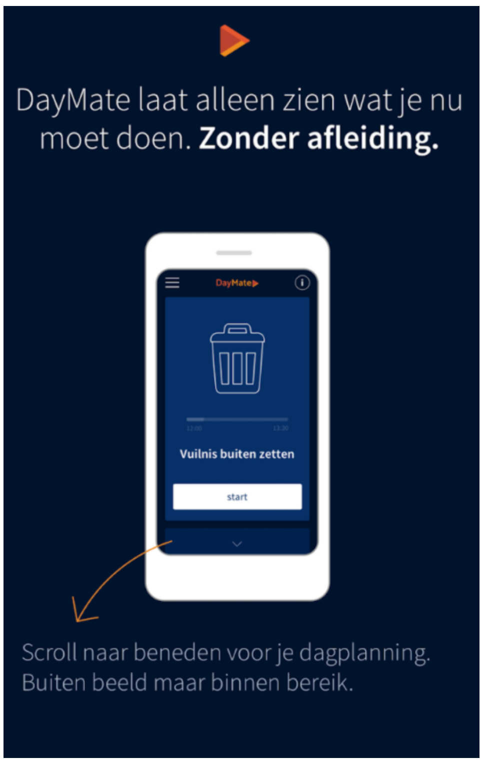
As mentioned in the introduction, the RitStick is a product envisioned by a student group as part of the “Designing for specific users” project for the Industrial Design Engineering bachelor. Multiple ideas were created through co-design with the target group, of the concepts developed, the RitStick was the preferred. The RitStick is a device that can be used in addition to the Ritme app. The concept of Ritme stays the same, users will get a notification and will move to the task location to turn off an alarm. The additional features that the RitSticks adds are the fact that the alarm will no longer come from the Ritme app. It will come from the RitStick. Instead of QR-codes, RFID chips are used. To turn off the alarm the RitStick needs to be held to the RFID chip corresponding to the location of the task. Apart from that, the project group found that people with autism often get stressed and anxious. To resolve this, a meditation function was added. A button was added that vibrates in the rhythm of a calm heartbeat and the RitStick screen will display breathing instructions.

2.3.3 Related planning tools

In this section, an overview will be given on planning tools that are related to the Ritme in some way. A description of the tool will be given and the takeaways for the Ritme will be listed. This can be seen in table 2.

Name	Image	Description	Takeaways
Calendars by Readdle [30]		Calendars is a smartphone app designed for iPhone and Ipad, which combines all other calendar apps into one, simple to use app. It also features task lists and reminders.	While this app is not specifically designed for neurodiverse people, it still has features such as reminders that aid in doing tasks. Therefore, using reminders can be considered.
Google Calendar [31]		Google Calendar is a calendar app, created by Google. It features all basic calendar features one could wish for. This includes reminders, repeating events and more. One additional useful feature is the integration with G-mail.	While this app is not specifically designed for neurodiverse people, it still has features such as reminders that aid in doing tasks. Therefore, using reminders can be considered.
Coach.me [32]		Coach me is a habit tracker app that uses community support as a way to motivate its users. Users set milestones, track their progress and receive virtual high fives from friends.	Community support can be a great tool in motivating users to do a task.

A physical calendar or agenda		A physical calendar can help with planning and keeping an overview of what needs to be done. In contrast to its digital counterpart, it does not offer any notifications, this is good and bad. Good because it offers fewer distractions, and bad because it cannot notify of tasks.	While a little old fashioned, the physical calendar is still useful. The overview aspect of this is its most important feature.
MijneigenPlan [33]		MijneigenPlan is a tool that helps caretakers of people with difficulty living alone create a plan. It consists of 3 parts: an online portal, an information board, and an app. The online portal is used to plan tasks and routines together, after which it is displayed in the app and on the information board.	One of the takeaways from MijneigenPlan is that giving an overview of incoming tasks is important. Furthermore, MijneigenPlan uses customisable colours, symbols and voices. This is something that can be considered for Ritme as well. Lastly, by including the caretaker MijneigenPlan makes it easier for people to plan tasks.
Krachtplanner [34]		The Krachtplanner is a planner specifically designed for people with autism, it is similar to a normal planner in many ways. The main differentiating factors being: it includes the ability to write down your	Even though the Krachtplanner is not a digital product, it has some features that can be useful to include in Ritme, for example, the relaxation methods.

		mood, the ability to create to-do lists, The ability to create a weekly goal, it includes space for reflection after every month, and it includes relaxation methods.	
DayMate [35]		DayMate is an app similar to a lot of calendar apps. The difference with this is that it is designed for people that need help with structuring their day. Instead of normal calendar apps which use lists, it uses icons. These tasks can then be split up into several smaller tasks. Other features that were found important by their user test are a “prikkelmeter” or stimuli gauge, which allows the user to show how their task went, and the ability to call their mentor/caretaker from the app.	Daymate was designed with the help of a team of experts relating to neurodiversity. Therefore features like using icons, caretaker integration and a reflection component are great things to consider using in the Ritme.

Clocky [36]		While not necessarily a planning tool. Clocky does demonstrate the effectiveness of a difficult to turn off alarm. When Clocky starts his alarm, he drives around. Requiring the user to chase him before the alarm stops. Therefore, the user is activated to get up, similar to how ritme works.	Clocky demonstrates that the more annoying an alarm is, the more motivation the user gets to turn it off. This should be carefully balanced with not having surprises, as found in section 2.1," Design for neurodiversity".

Table 2 -Planning tools that can inspire future developments of Ritme

2.4 Requirements and guidelines

Based on the research done in the state of the art, several requirements can be identified for the final prototype of this research. These requirements are not final as during later phases, new requirements might appear or current requirements may become redundant. In this section, an overview of the requirements is given. Furthermore, guidelines are identified that serve as a goal for the Ritme, these are not vital to Ritme's functioning but may improve user experience. Finally, some ideas for features to include are written down based on other products and the literature reviewed. Figure 15 shows where the requirements and guidelines fit in.

2.4.1 Requirements

1. The software of the end product should be customisable to fit the user's needs. [13]–[18]
2. Stimuli should not be overdone, be consistent and careful consideration should be taken if they are necessary. [16]
3. Information should be displayed clear and objectively [14], [15], [19], [21]
4. The layout of the software should be calm [21]

2.4.2 Guidelines

1. Digital elements should be mapped to physical ones, and the other way around [23]
2. Intangible aspects should be used to complement tangible ones [23]
3. Interactions with tangible aspects should be based on common knowledge [23]
4. There should be a clear continuity between tangible and intangible parts (isomorph effect) [22], [23], [26]
5. Input should be space-multiplexed[22]
6. Lightweight interaction should accompany the main interaction [22]
7. Tangible components should be relevant [22]
8. Tangible components should be expressive [22]
9. The product should try to contain all six types of unity between action and reaction [26]
    a. Time
    b. Location
    c. Direction
    d. Dynamics
    e. Modality
    f. Expression

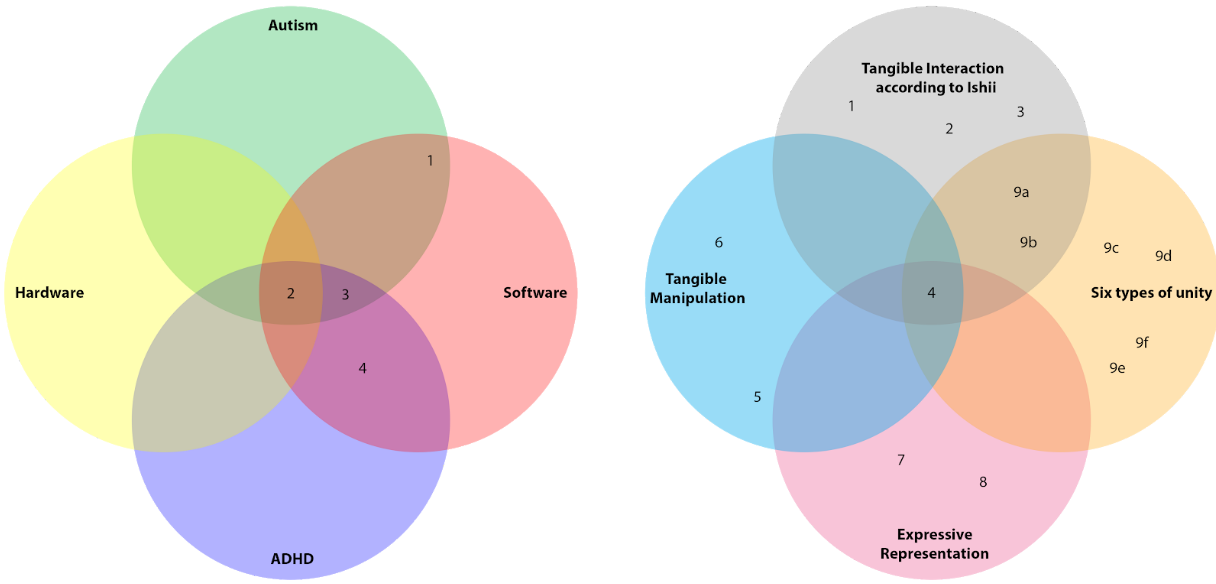


Figure 15 - Euler graphs of the requirements and guidelines

2.4.3 Other ideas for Ritme

The following ideas were found throughout the state of the art and may serve as inspiration for future developments of the Ritme app. These ideas will not be used in the current research as they do not fit well within its context of designing a tangible device.

10. Community support can be a great motivator [32]
11. Having quick access to, or allowing the caretaker to see progress is appreciated. [33], [35]
12. Relaxation and reflection methods can be included [34], [35]
13. Allowing users to get an overview of their tasks is appreciated [30]–[34]
14. Positive reinforcement can be used to stimulate continued use [21]
15. Markers can be used to show the progress of the user [21]

2.4.3 Design vision

The final product developed should at a minimum be in accordance with the requirements found, furthermore, the guidelines found should be taken into account as they provide a basis for the tangible user interface that is to be developed, however, it is not necessary for all of these to be achieved. However, as nice as requirements might be, they do not explain the entire picture. In the end, the most important factor is that the stakeholders are satisfied, therefore, user tests will be conducted and stakeholders will be involved during the design process.

The end product will be inspired by a combination of multiple products found in the state of the art. For example, the cognitive switching aspect of the Clocky [36] will be used and the tangible interaction will be inspired by the Microsoft surface dial [28] and the Siftables [29], as these are both handheld devices including tangible interaction. While interesting, Sandscape's [27] interaction method won't be used as it is not feasible in this context. Furthermore, the different planning apps won't be involved in the design process of the tangible device as Ritme is already a functioning planning app and improving that is not within the scope of this research. However, the information found may be useful in future research. Lastly, the Ritstick will not be used as inspiration. This is because doing so might negatively influence the creativity in this research.

3. Method

3.1 Design process

The design process used in this research is the “Design Process for Creative Technology” created by Mader and Eggink [37]. This design process consists of 4 main phases. Ideation, Specification, Realisation and Evaluation. Within each of these phases, there is a high focus on iteration. However, before these 4 design phases can start, a state of the art research was done as inspiration for, and to gather requirements for the ideation phase. Finally, when the evaluation of the prototype is deemed sufficient, the concept is finished. A general overview of the design process can be found in figure 16.

3.1.1 Ideation

The ideation phase can start at different points, it may be a creative idea, technology or user needs, but in this project, the starting point is the stakeholder’s requirements. Mader and Eggink say: “we share the conviction that creativity can be trained to a certain level, and is more often the result of hard work than the kiss of the muse” [37]. Therefore, the ideation phase consists of using divergence and convergence techniques to come up with creative ideas. Other inspiration may be related works, one can combine this with original ideas to form a new idea which possibly is even better. During this phase, brainstorms will be done (diverging), after which these ideas will be combined (converging) to come to several elaborated project ideas. These ideas will then be discussed with the project supervisor and client after which this feedback will be used in the next brainstorming session.

3.1.2 Specification

The specification phase uses the ideas found in the ideation phase. In this phase, the rough ideas formulated in the ideation phase will be further refined. This is done by creating scenarios, storyboards, or quick prototypes. These will be shown to and evaluated by users and stakeholders. After these evaluation sessions, the prototypes will be refined and evaluated in a second evaluation session. Finally, the results of this evaluation can be used in the next phase.

3.1.3 Realisation

During the realisation phase, the final prototype will be built based on the requirements found in the specification phase. This phase consists of 4 different sub-phases; Decomposition, Realization of components, integration and evaluation.

The decomposition phase is the analysis of components needed, after which realisation of components can begin by buying and ultimately combining them. The next step, evaluation, consists of self-evaluation of whether or not the requirements were fulfilled.

3.1.4 Evaluation

In the evaluation phase, the success of the final prototype will be assessed. Whether the requirements for the prototype will be met will be evaluated through user testing. Furthermore, a critical reflection will be done on the design decisions. If everything is deemed sufficient, the design process will end, if further improvements are still needed, the design process will continue with the ideation phase.

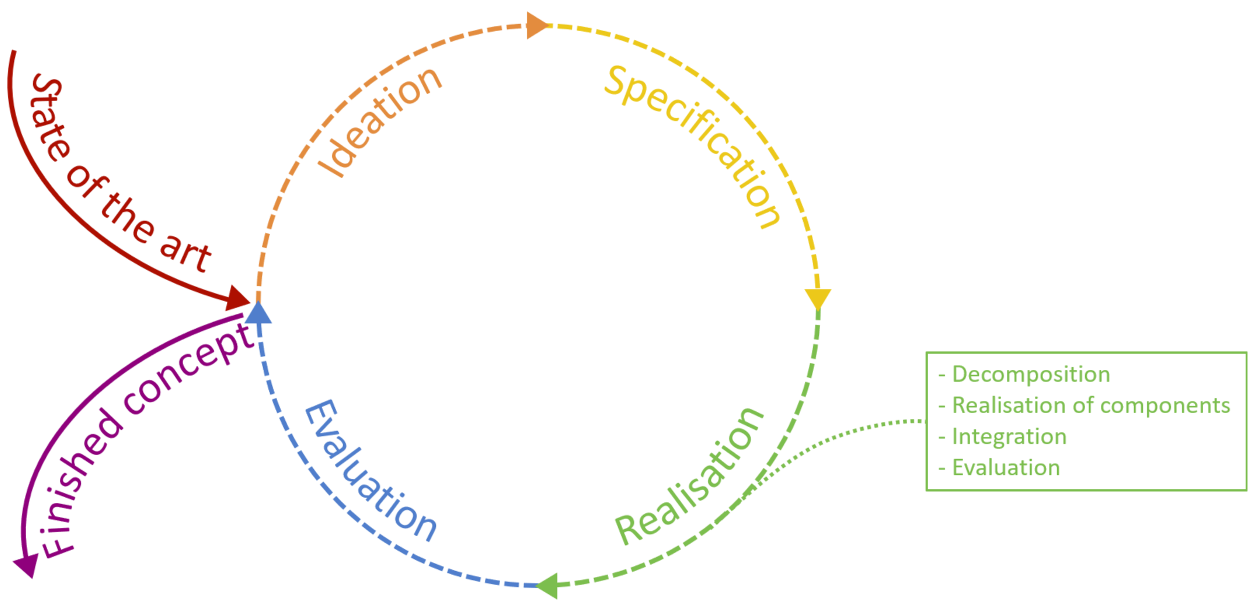


Figure 16 - Overview of the design process for Creative Technology

4. Ideation

During the ideation phase, a similar process to “object brainstorming” was used. This method was chosen because creative types of tangible interaction are necessary for the final product. Therefore, the classic ways of brainstorming involving writing down ideas, for example through the use of a mindmap, were less applicable here. These methods would hinder creativity by requiring the interaction to be described in words. The brainstorming process consisted of 4 phases.

First, all functionalities that the eventual product may have were written down. And dissected into more basic interactions. See figure 17. This was done to give a clear overview of the system which allowed me to easily couple the interactions to the functions.

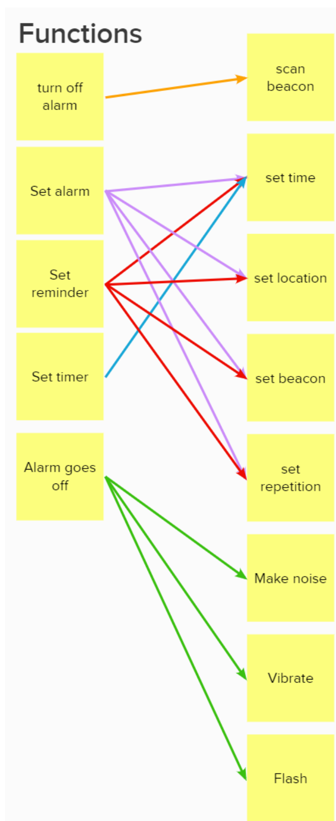


Figure 17 - Functions of the system split into their general parts

After this, various objects were collected and fumbled with. These objects were collected from around the house and from the Gamma hardware store. If an interesting type of interaction occurred this was filmed and placed into the grid of interaction. See figure 18. This is the divergence phase of the brainstorm. From this grid interactions were chosen that may apply to a certain function. For example, for the “scan beacon” function, it was known that two components needed to come together, namely, the beacon and the scanner. Therefore, all interactions which included two components interacting with each other were added to this function. See figure 19.

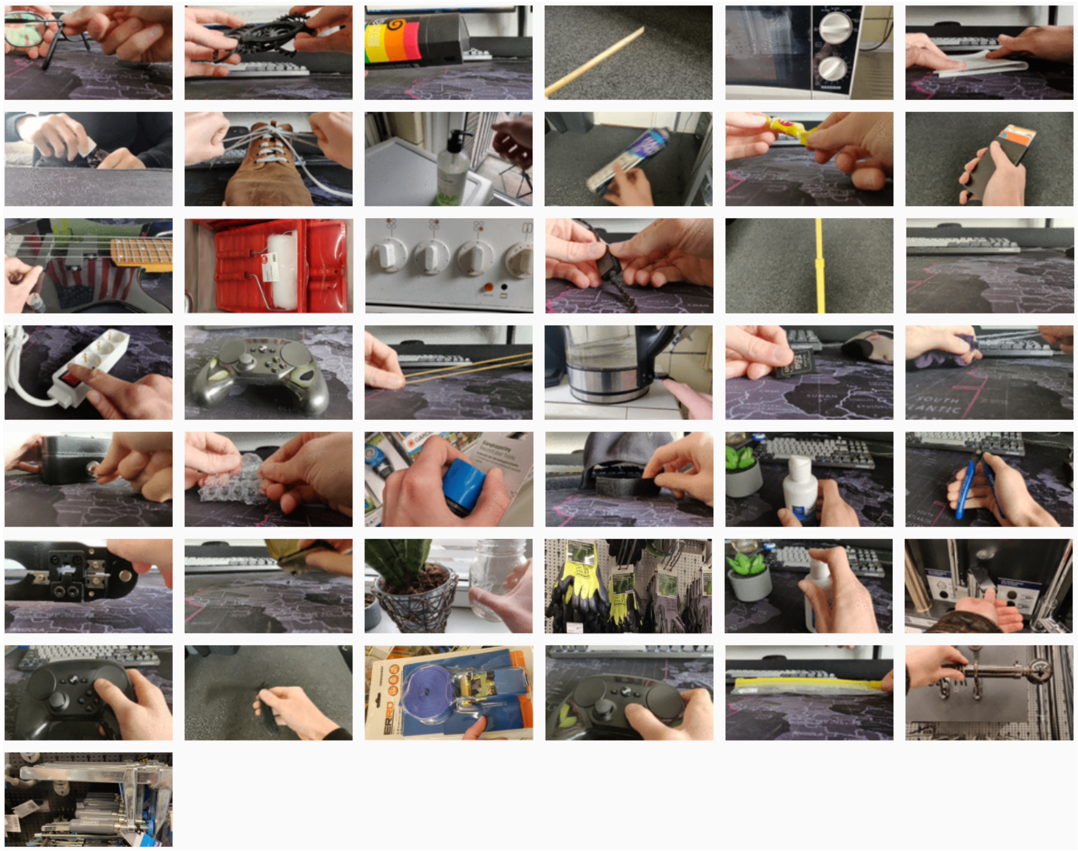


Figure 18- Interaction grid containing short videos or photos of all unique interactions found

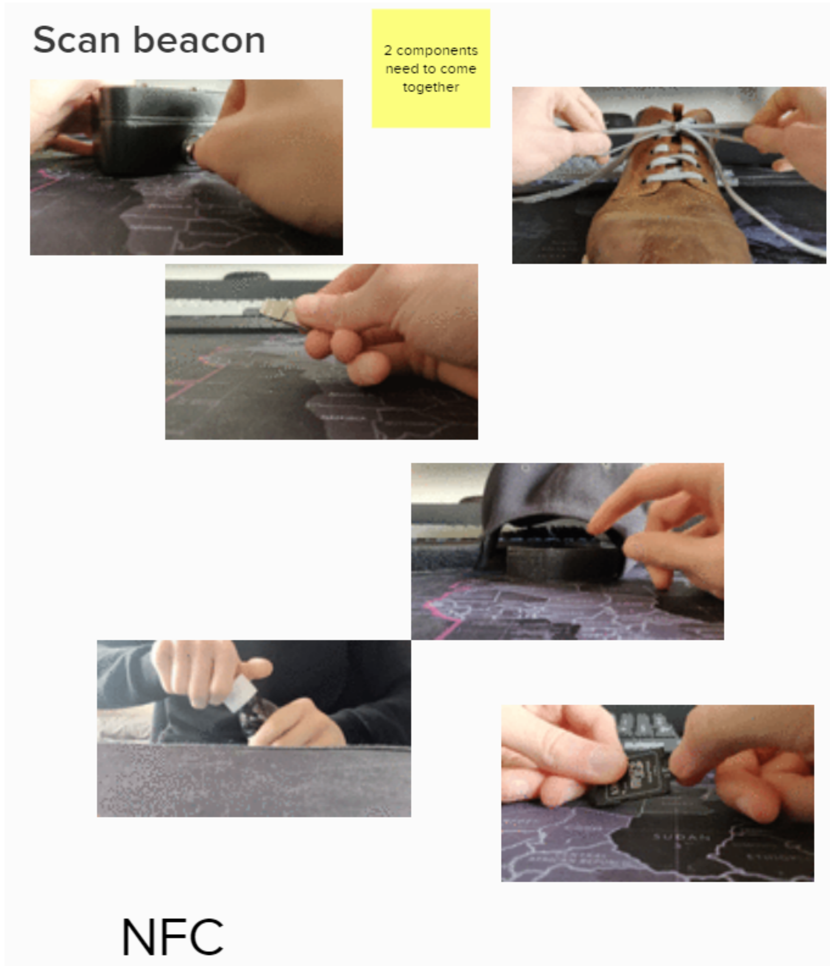


Figure 19 - Interactions corresponding to the “scan beacon” function

Finally, ideas for the final product were generated through the use of the function sections. One interaction method was chosen to be central and other interaction methods were added to form new ideas. Ideas were generated with all functions, with just one, and with some functions but not all. For example, idea 8 includes the functions “turn off alarm” and “set time”, while not including other functions like “set repetition”. An overview of the generated ideas can be found in figure 20. The specification phase will determine which of these ideas is preferred by stakeholders and how to improve upon them. Along with that, my general thoughts about these prototypes can be found in figure 21. The final mural can be found in Appendix 1.

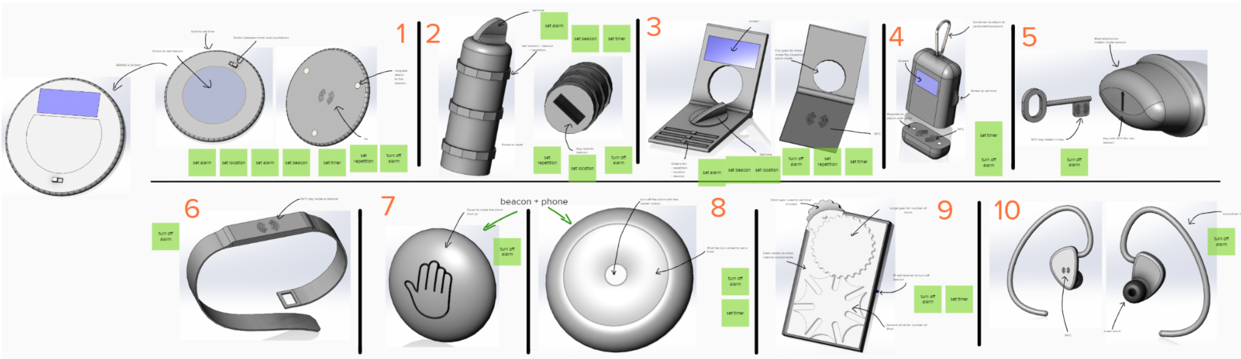


Figure 20 - An overview of the ideas generated during the ideation phase

Things I like

- Knocking to set things (1)
- magnets to connect things (1 + 4)
-rotating to set time (1 + 2 + 3 + 4 + 8 + 9)
-simplicity of having beacons + phone (7+ 8)
- visual feedback without a screen (2+9)

Things I dislike

- higher difficulty when putting electronics in beacon instead of intermediary device (5 + 6 + 7 + 8)
- clunky design of 2
- non intuitive design of 1
- a lot of space for just 1 feature (1 + 9)

Figure 21 - General thoughts about the ideas generated

5. Specification

During the specification phase, the ideas generated in the Ideation phase were refined until one final idea is generated (convergence). This phase started with the creation of a survey. This survey was sent to 3 potential users that are diagnosed with ASD. These participants were found through the AssortiMens foundation. Furthermore, Alderick (the owner of Ritme) was consulted on his opinions of the generated ideas.

5.1 Feedback from Alderick

During an online meeting with Alderick, his feedback was asked on the 10 ideas generated during the ideation process (see figure 20).

Alderick was quite short about both ideas one and two. He liked the idea, however, the form factor is not ideal to be carried around. Which is one of the general feedback points he gave. It should be natural to carry around the device. As in that way, it solves the issue of people not carrying their phones around with them all the time. Idea 3 was received more positively, Alderick stated that he liked the feature of being able to quickly set an alarm using the big dial and he liked that the device could fit in one's pocket easily.

For ideas 4 and 5, Alderick liked the simplicity. The device did not need to replace all functions that the Ritme app currently has. Idea 5 further received the feedback that the idea of a key is nice in terms of tangible interaction, however, because the item will be created out of plastic, it is quite fragile, therefore, another, less fragile shape would be preferred. During the discussion of idea 5, another preference was found. We discussed whether the beacons should have features like being able to quickly set an alarm. The conclusion from this discussion was that the beacon should be as simple as possible. The more features the beacon has, the more expensive it is to make and therefore buy. Users will need multiple beacons, therefore, keeping the price low is vital. Alderick used the Apple AirTags [38], as an example of what should not be done. The Apple AirTags can be attached to personal belongings which can then be found using the app. Similar to how the device to be developed for ritme connects to the app. However, each AirTag costs €35, which is quite expensive if multiple tags are needed.

Idea 6 was quickly put down as Alderick did not want to compete with other smartwatch devices as of yet, even though he did think it was an idea to further develop this in the future. Idea 7 and 8 both do not include an intermediary device. This was disliked by Alderick as it could occur that the user does not hear the alarm.

Idea 9 is mostly focused on the look of the final product. Although Alderick did like the idea of seeing the internal components, it is still too early in the development process to address this. Therefore, focus should go to other aspects of the final product.

Finally, idea 10 was evaluated. Since people like to wear their own headsets, this idea was not supported by Alderick. He mentioned that he even has 2 headphones that he uses interchangeably. Therefore, having another device that is supposed to go on your ears is not ideal.

In conclusion, new requirements for the device have appeared. This being that the product should be natural to carry around, the beacon should not contain a lot of electronics and along with that the new device should not be able to do everything the Ritme app currently does, the beauty in simplicity should be used.

5.2 Online survey 1

In parallel to getting feedback from Alderick, an online survey was created. (See Appendix 2). For this users were found that would be representable for the final users of the Ritme, namely, people with ASD (for ethical approval refer to reference number RP 2021-44). Sadly, only one out of three participants filled out the survey. However, some useful information was still collected.

As with the feedback from Alderick, the participant liked the simplicity of idea 5. The participant states: “A key is clearer than a QR-code you have to scan (I often fail at doing these kinds of things)” (translated from Dutch to English). Therefore, for the final design, more importance is placed on the ease of use of the device.

For most ideas, the participant was worried about how the device would be carried around. This is understandable as most generated ideas did not show a clear indication of how to carry them.

This further confirms the requirement for a portable device. For that reason, I have decided to include multiple carry options in the final design.

5.3 Further design iterations

Following the meeting with Alderick and the answers from the online questionnaire, I decided to continue working on idea 5. The focus of the redesign was mostly on creating a more sturdy design as compared to the key, furthermore, simplicity should be maintained. Figure 22 shows the designs that came out of that.

Figure 22 - Results of the second ideation session

The “key device” (bottom left) is most similar to how idea 5 worked. It functions in the same way as a key in the way that you put it inside the beacon and turn it. While working on the idea in the centre, I got inspired by the shape of a hip flask. Combining my idea with the shape of a hip flask resulted in the idea on the right. I discussed all these with Alderick and he liked this last idea the best. Due to its shape, it is easy to carry around and can only be inserted into the beacon one way, allowing for simple operation. This is still similar to how a key can be inserted only one way, however, the action of turning the key has been removed.

5.4 Technological prototype

After choosing the general shape of the product, a technical prototype was created using my 3D printer. This was done to show that the concept is technically feasible to create. This prototype can be seen in figure 23. For the user the prototype works in the following way:

1. The user will set an alarm on their phone. They will specify the task place and can customize the alarm sound, vibration, and colour.
2. The alarm will activate on the physical device
    a. The device will light up, vibrate, and make sound depending on the chosen location (for example, lighting up green, and vibrating and playing the Mario theme)
3. The user will take the physical device to the task location and place it on the beacon, after which the alarm will turn off.

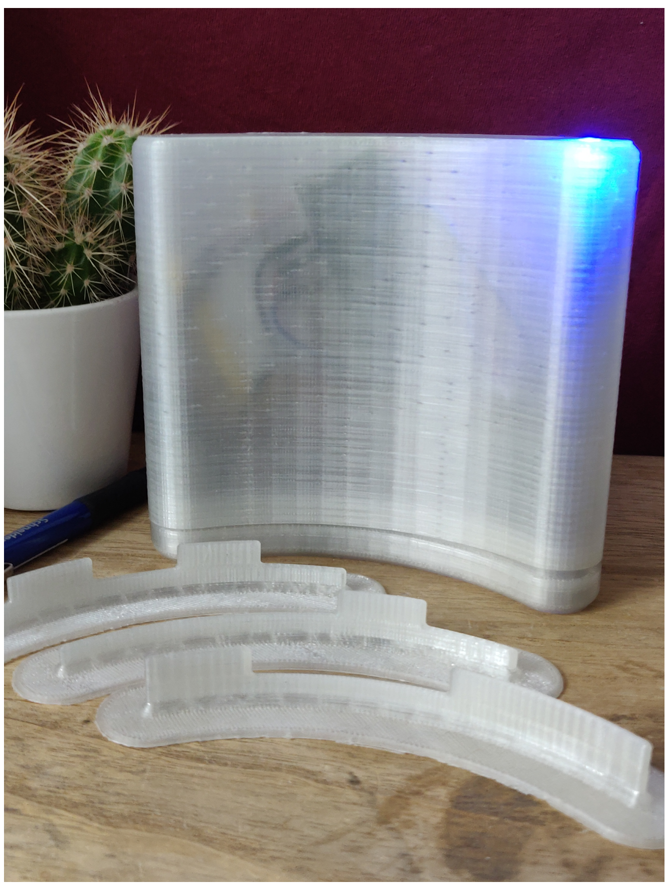


Figure 23 - Final technical prototype. The main device sounding the alarm (as indicated by the blue LED), and three beacons in front.

This shape was chosen as it fits well within the pocket of the user (it conforms to the shape of the leg) and it can only be inserted into the beacon in one way. It should be mentioned that this device is on a 2x scale, since that way the prototyping electronics fit. This is mostly caused by the use of a microcontroller, which is quite significant in size. The device will detect whether or not the right beacon is inserted by utilizing 5 limit switches placed along the opening of the physical device. A beacon has 5 sections which can be high or low. With this, the device can read whether one of these sections is high or low, ending up with a 5-bit signal. This makes for a total of 31 options. The option of having all beacon sections low is not counted as in that way the device does not know whether a beacon is inserted. This method of beacon detection was chosen over other easier methods like NFC because of ease of creation and costs. For the device to work to its full potential the user would need to have multiple beacons. If the beacons would include any electronic parts this could quickly become expensive to buy. Furthermore, implementing electronics during the manufacturing process is quite difficult. Further establishing that having a beacon as simple as possible is preferred.

The communication between phone and device is done through Bluetooth, specifically Bluetooth low energy (BLE). This was chosen because the device is battery-powered, and Bluetooth low energy consumes less power, due to the fact that the Bluetooth will not be enabled unless called. In this case, the device works as a BLE server with which the smartphone can connect. The alarm protocol works in the following way.

1. The user sets a time with a corresponding location, colour, sound, and vibration pattern in the Ritme app.
2. When it is time for the alarm to sound, the app will send this information to the physical device via BLE.
3. The physical device will start activating the buzzer, vibration motor, and LED so the user knows what location to go to.
4. If the device is placed on a beacon, it will scan if all correct limit switches are activated and turn off the alarm if this is the case.

When it is time for the alarm to start, the Ritme app will send a code with all the needed information to the device. It was chosen that the app does this as in this way this information is stored on the user's phone, resulting in saving memory, computational energy, and battery of the device. The ritme app will send something similar to "0,1,1,0,0,255,000,255,3" figure 24 shows how this can be interpreted.

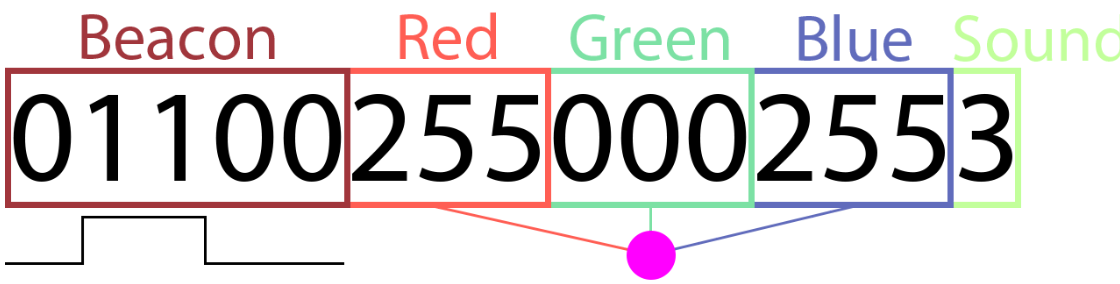


Figure 24 - Components of the signal sent to the device from the app

5.4.1 Prototype creation

The process of creating the prototype started with sorting the system's functions by priority, after which options were chosen for how to achieve these functionalities. These options were then evaluated on feasibility among other factors. For the scanning functionality, object detection was chosen as it allows the beacons to only consist of plastic. This results in beacons that are very cheap to manufacture and therefore also cheap to buy for the consumer. Which was an important thing to consider as mentioned before. One component that might be expected but isn't included is a real-time clock (RTC). An RTC can be used to keep track of time even when the system is turned off, which would seem necessary for an alarm. However, in this case, an RTC is not necessary as the phone of the user will handle this. The process of linking functionality to ways in which this can be achieved is shown in figure 25.

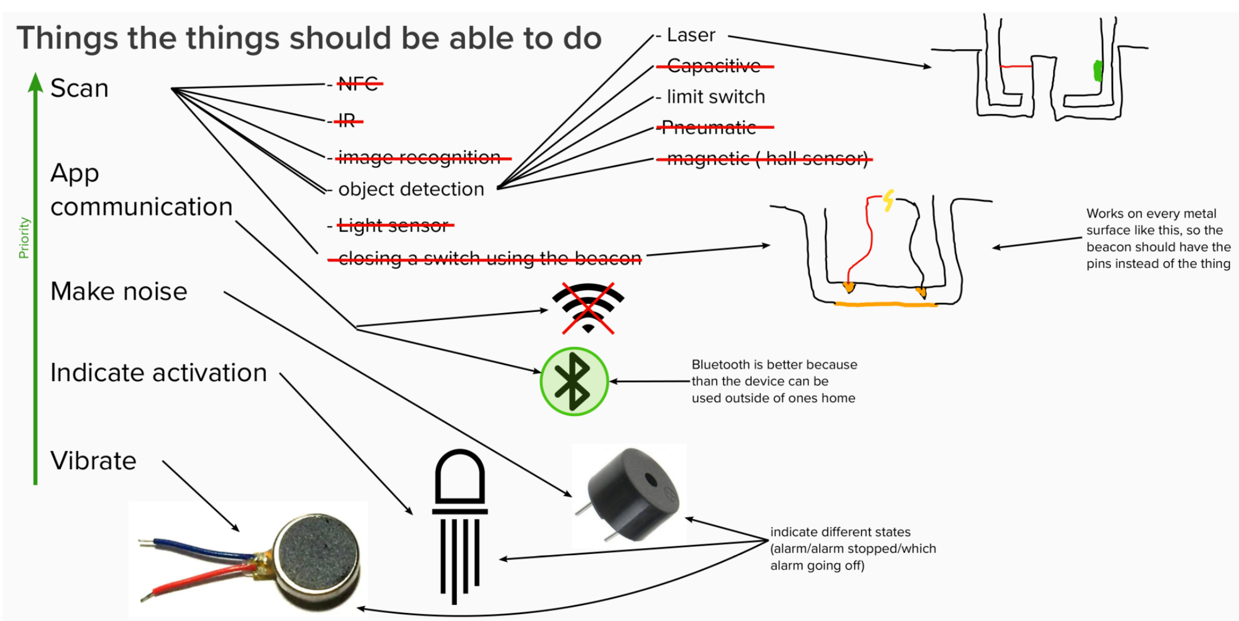


Figure 25 - Linking between system functionalities and possible ways to achieve these

After choosing how to achieve all the functionalities, the components for this were chosen. An overview of this can be found in figure 26. First, easier components were chosen. For example, a common cathode RGB LED was chosen to serve as RGB LED. After these components were chosen, the pin requirements for them were specified and totalled. With this, a choice was made for the microcontroller. Apart from the pin requirements the microcontroller also needed to have a Li-Po battery charger and connection, and Bluetooth functionality. For this reason, the WEMOS lolin32 was chosen.

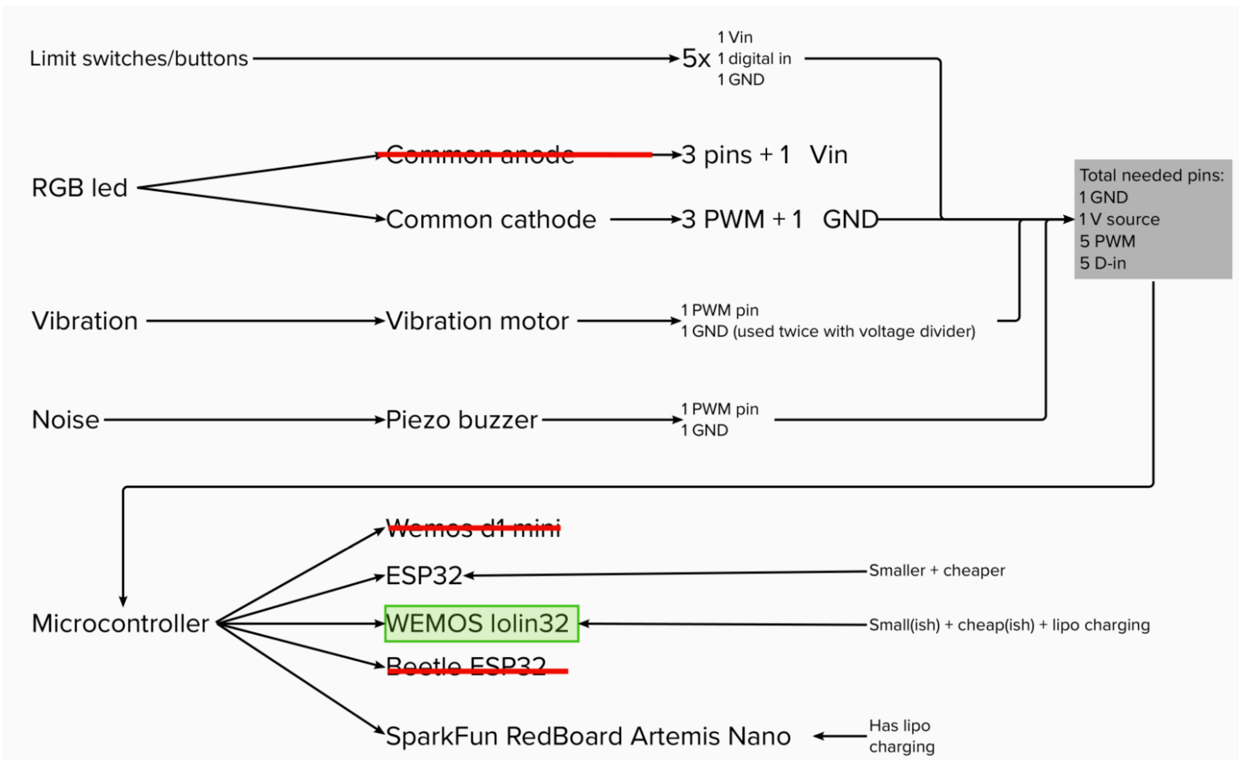


Figure 26 - method of choosing components. First finding the necessary pins needed, then choosing the best microcontroller.

Along with choosing components, the 3D model was refined and made to fit the ordered components. The final result of this can be found in figure 27. The final technical prototype created from this 3d model can be found in figure 23. As can be seen, the prototype consists of the components specified in figure 26. These components were connected to the microcontroller following the circuit diagram that can be found in figure 28. At a voltage of 3.3 volts, the manufacturer of the RGB LED recommended a series resistor of 82Ω in series with the red LED [REF]. This ohmage value was achieved by combining 4 330Ω resistors which were available, see equation 1. Finally, the programming of the microcontroller was done through the Arduino IDE. The code for this can be found in Appendix 4.

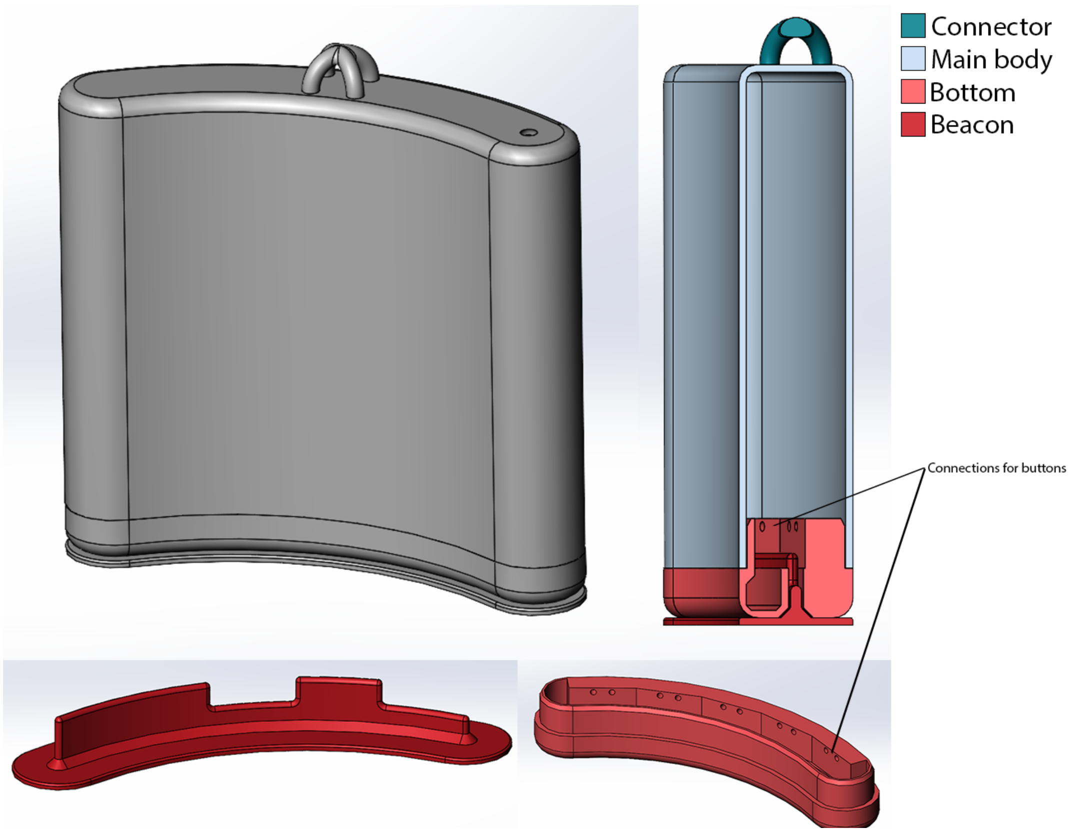


Figure 27 - 3D model of the technical prototype.

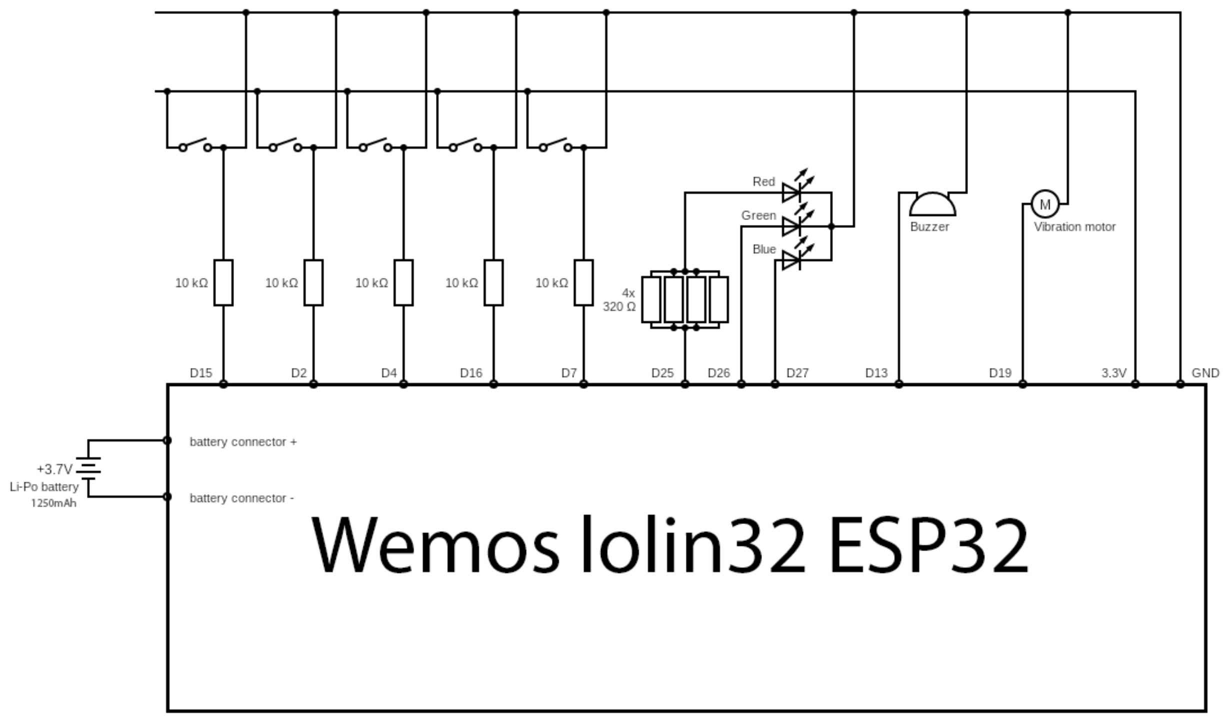


Figure 28 - Circuit diagram of the technical prototype

$$R = \left(\frac{1}{\frac{1}{330} * 4}\right) = 82.5\,\Omega$$

Equation 1 - Calculation of the series resistor for the red LED

5.5 Online survey 2

Apart from tangible interaction being about physically doing the thing you are supposed to do (for example pushing a button), it is also about the psychological effect that an interaction has on a person. An example of this is the coat rack. You arrive home after a long day of work and the first thing you do when you get home is hang up your coat on the coat rack. Apart from serving as a functional item on which you hang your coat, the interaction with it also signifies the end of the workday. The technological prototype is lacking in this psychological aspect. For that reason, a second online questionnaire was conducted, which can be found in appendix 3. The participants were shown 5 interactions and were asked to choose which of these interactions they liked the most for:

1. Turning off the alarm of Ritme.

2. Notifying the Ritme app that you are done with your task.

In hindsight, it would have been better to not notify the participants this was for Ritme, and instead create questions like: “what interaction do you associate most with starting a task” and “what interaction do you associate most with being finished with a task”. For this reason, some responses did not fit with the original intent of the questionnaire, to see what people’s intuition is for the interactions with Ritme. After filtering the responses that were based on intuition the result is as follows. For turning the alarm off the option of holding the device to the beacon (like using an NFC tag) was preferred, for notifying the app (and yourself) that the task is done the option of holding the device to something was also preferred. To me, this seems illogical as this does not truly reflect starting a task or finishing the task. Rather they seem to be related to other interactions that happen often in daily life. For example, when using Apple Pay to pay for something you are holding a device to something else. One of the ways that tangible interaction works is by mimicking real-life actions and coupling these to digital inputs. As with the video recorder seen in figure 8. In contrast, in this case, the actions performed are actions that occur in life regularly, they are not directly linked to the input to the device. Don Norman calls this a cultural constraint. “Cultural constraints are conventions shared by a cultural group.”[39] The ability to use NFC or RFID in daily life has become so prevalent that it has become one of the options people consider to be within a product's abilities when no other usability can be found. For that reason, participants may have found this option the most intuitive. The action of holding the device to the beacon was not considered further for this reason, as it did not fit well with the intended purpose of creating a psychological effect for the user. However, it is interesting nonetheless.

5.6 Interaction prototype

To gather more information about the preferred interaction a brainstorming session was held with 2 fellow students. This brainstorming session started with the questions: “what action signifies starting a task to you?” and “what action signifies being finished with a task to you?”. These questions were chosen to not limit the thought process to the Ritme device itself. Later in the process, the answers to the questions were coupled to the question “which of these interactions can be added to the technical prototype, while not completely redesigning the device?”. After having answered these questions, the participants were asked to choose their favourite options, both for interaction and a celebration. The number 1 favourites were coloured dark green, while runners ups were made light green. One of the cards was also made dark blue, representing that the concept was interesting however it would most likely not be feasible.

The outcome of this brainstorming session can be found in figure 29, a larger version can also be found in Appendix 5.

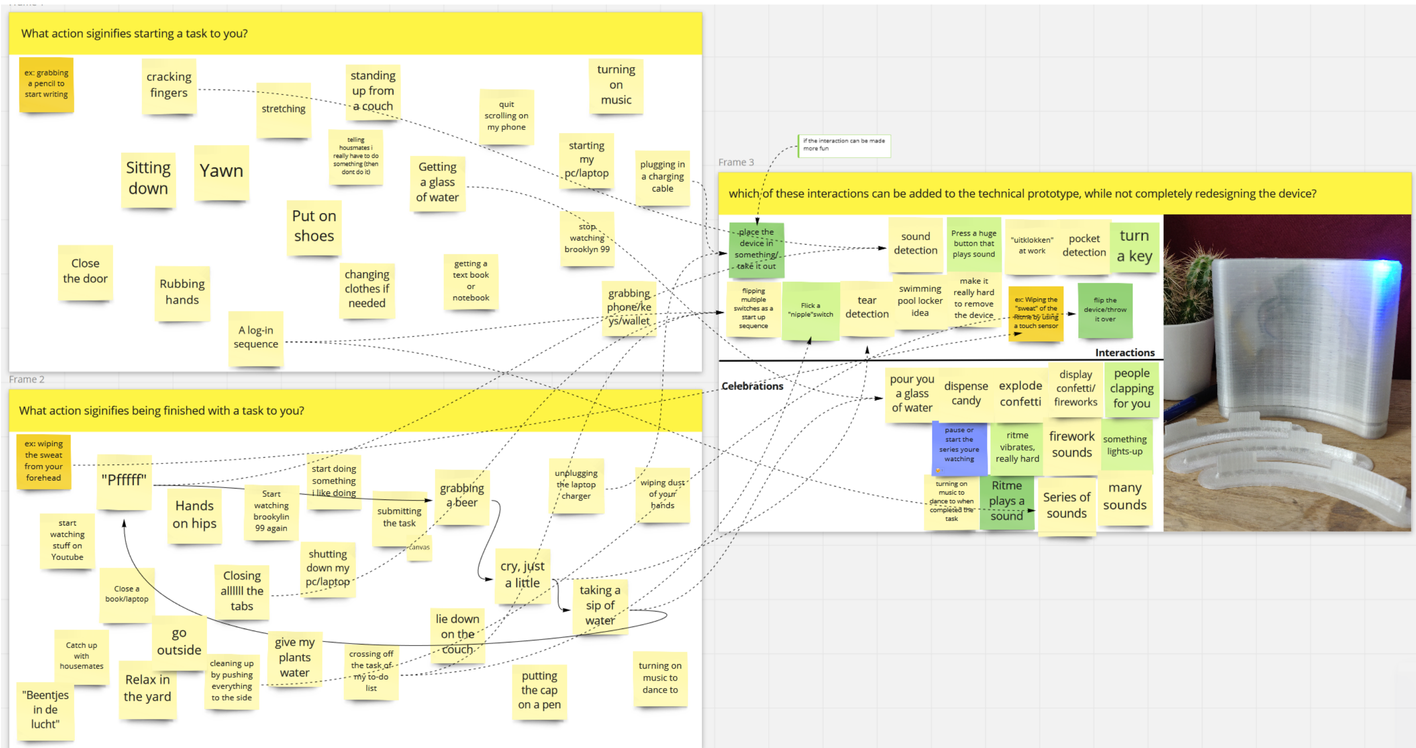


Figure 29 - Outcome of the interaction brainstorming session

The most favoured interactions were: “place the device in something/take it out”, and “flip the device/knock it over”. The first of these is very easy to implement into the current design of the technical prototype. Instead of making the beacon have flat edges, these edges can be made to allow the device to slide into the beacon. The action of flipping the device for it to activate is harder to implement into the current design as it does not include a sensor that can detect a flip, e.g. an accelerometer. One way to overcome this is to have a default position for the device and have the interaction occur when the device is flipped. This can be done by using affordances as described by Don Norman [39]. By removing the affordance of placing the device upside down without a beacon, the default position can be set. And by placing the slot with limit switches on that same side, the perceived affordance will be that the user must flip the device before it can be inserted into the beacon. The iterations of how these versions were developed can be found in figure 30

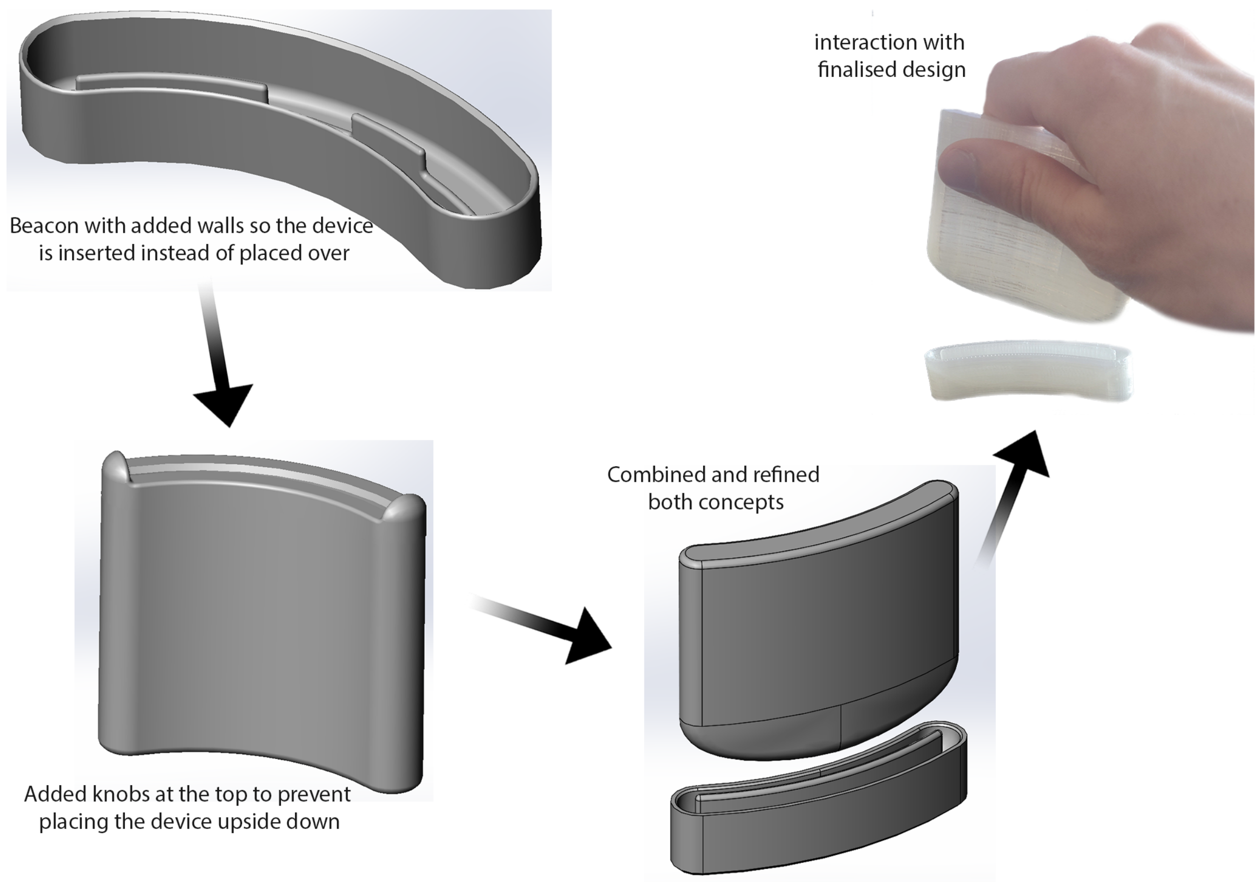


Figure 30 - Iterations of the interaction prototype based on the outcome of the brainstorming session.

5.6.1 User experience

After the initial setup of the Ritme, the user experience can be described in six steps.

1. The Ritme device notifies the user that a task has to be completed.
2. The user takes the Ritme to the task
3. The user places the Ritme device on the beacon
4. The user does his task
5. The user takes the Ritme of the beacon
6. The user takes the Ritme device with him ready for the next task

An overview of this can be found in figure 31. Steps 3 and 5 could signify more to the user than just turning off the alarm. While this is of course up to the user's interpretation, the action of taking the alarm out of the beacon may signify that the user is done with the task, therefore, the user may not allow themselves to take the device or leave the task location without actually finishing the task. This may serve as some extra external motivation for the user.

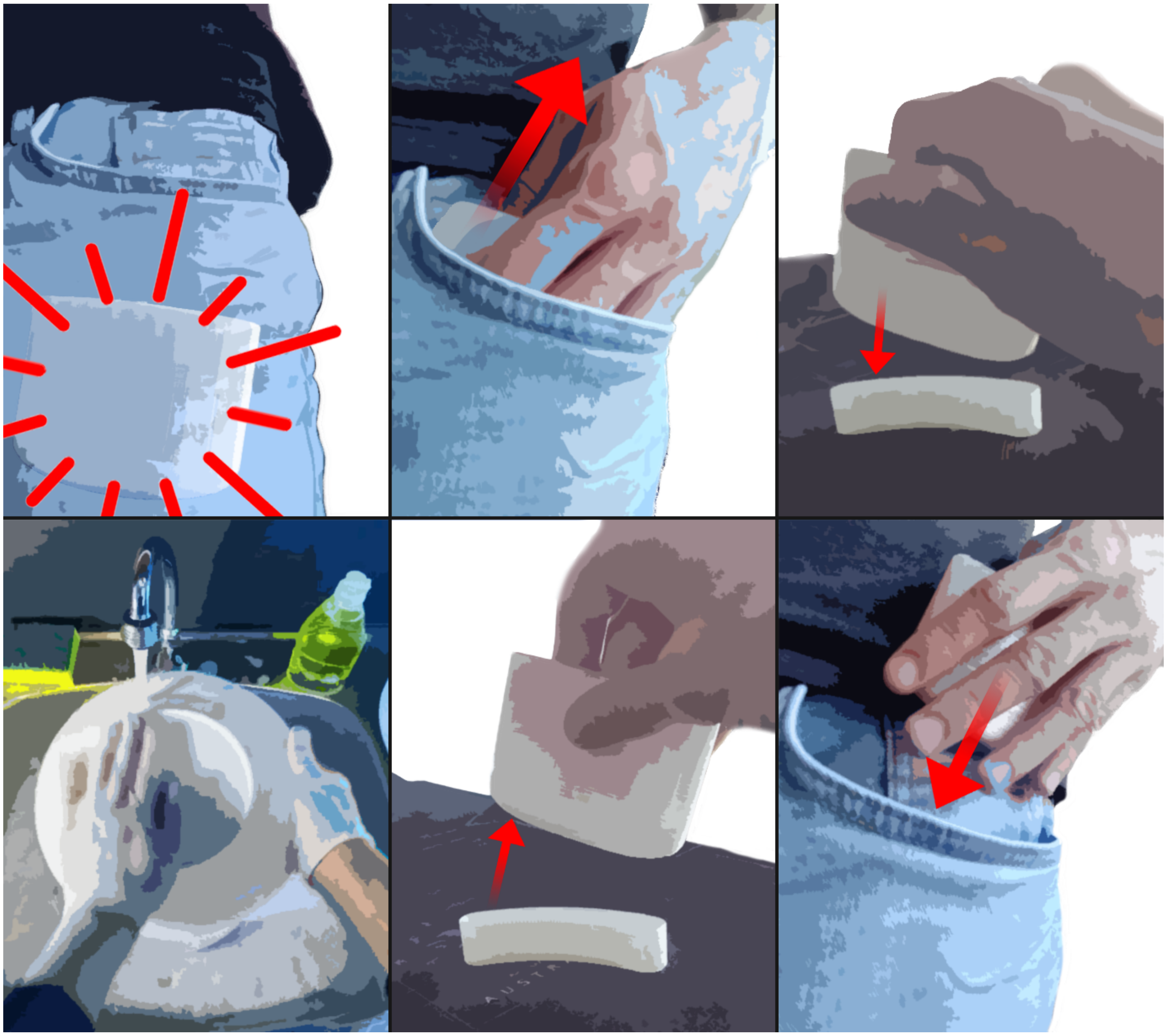


Figure 31 - An overview of the user experience of the new Ritme device.

5.6.1.1 Portability

While not integrated into the current physical model of the device. During the ideation phase, it became apparent that it would be good for users to have multiple ways in which they could carry the device. The current physical prototype is focused on being carried in the user's pocket, however, by integrating a small connection point (as can be seen in figure 27), possibilities open up for the user. By including several items in the product's box the user can choose which method of transportation they prefer. Some possibilities include:

- Including a keyring → the user can connect the device to their set of keys
- Including a carabiner → the user can connect the device to the belt loops of their pants

- Including a necklace → the user can wear the device around their neck, making it possible to easily transport it without the necessity of wearing pants (for example, for women wearing skirts)

Including these options will give the user flexibility to carry the device however they want. If this was not the case, the user experience would decrease drastically as having the alarm go off without having the device with you would result in having to walk to the device first, before walking to the task location.

5.6.1.2 Feedback and feedforward

The feedback the user receives from the device is still similar to the technical prototype. The most obvious kind of feedback present is the functional feedback of the Ritme. When you place the device in the beacon, the alarm shuts off. However, augmented and inherent feedback (as discussed in sessions 2.2.3.1) are also present. Augmented feedback is present when the alarm is turned on. The device will vibrate, it will blink a certain colour, and it will play a melody. These actions all indicate to the user that the alarm is currently on. More subtle feedback is also present in the form of inherent feedback. When the user pushes the device into the beacon, they feel the interaction between the device and the beacon, furthermore, the clicking of the buttons when inserting the device can be heard and felt, which may be considered satisfying by some people. In this way, two feedback loops are implemented. The first feedback loop is present through the inherent feedback of having a physical device. While the second feedback loop is present through the combination of functional and augmented feedback.

The newly created device is not lacking in inherent feedforward (also called expressive representation by [22]). The shape of the device being the same as the beacon clearly shows the connection between them. Furthermore, the shape of the device prevents the device from being inserted into the beacon in the wrong orientation. This same principle also applies to how the device conforms to the user's leg when carried in a pocket. Therefore, showing that a good method of carrying the device is in the user's pocket. However, the new device is lacking in terms of functional and augmented feedforward. The shape of the new device does not explicitly show the function of the device. It is hard to show the functionality of "being an alarm which can be turned off by being inserted into a beacon". This problem can be overcome by the use of augmented feedforward. In this way, the augmented feedforward makes up for the lack of functional feedforward. This can for example be done through a manual or labels on the device.

6. Evaluation

The evaluation of both the technical and interaction prototype were both performed with Alderick. This is because the prototype is very early in its development, therefore, inviting end-users is not appropriate yet. Alderick was asked to perform the entire user experience with the new Ritme device. First, he was asked to set up the alarm. This was done through a paper prototype instead of through the ritme app, as this app does not include the appropriate features for the new device yet. Afterwards, the technical prototype was activated and Alderick was asked to turn off the alarm, without any further guidance. While doing so he was asked to commentate on any of the actions he performed, the so-called “think-aloud protocol”. After doing so he was asked questions related to the technical prototype. After answering these questions, he was introduced to the interaction prototype and further questions relating to this were asked. The full protocol of this and Alderick’s answers can be found in Appendix 6.

During the setup of the device using the Ritme prototype, Alderick already noticed one thing. Alderick decided to use the beacon with all sections high, this caused some confusion when setting up as it was unclear whether the sections were high or low. This is something that the app could solve easily, for example, by not allowing the user to set up an alarm with which all beacon sections are low (because this beacon also isn’t possible). Overall, Alderick was pretty enthusiastic about the prototypes and after implementing some changes he could see him continuing to work on this.

One of the suggestions was on the types of functionalities the physical device would take over from the mobile phone application. The current prototype is focused on increasing the lifespan of the physical device, something alderick also mentioned is important. Therefore, the device only has the capability of sounding the alarm and turning off the alarm. Several Ritme users mentioned that, due to the use of GPS signals, their phones would drain much faster when using Ritme, therefore, Alderick stated that it is important to take away as many power-draining features as possible from the phone and implement them into the physical device. This could also improve the accuracy of the GPS signal if a better sensor is used or if the GPS location is combined with wifi location. This is the biggest shift of focus that was discussed during the evaluation.

Another issue that was addressed was the possibility of adding a dismiss alarm button. This is useful in case the device malfunctions or the user loses one of the beacons. Apart from this, it could also be useful in cases where the user does not feel like doing the task. However, this is also a downfall of adding a dismiss alarm button. If it is very easy for the user to avoid doing the task, the main goal of the Ritme is not met. One solution would be to include a dismissal button with a timer that requires the user to hold a button before the alarm is dismissed. In this way, the user is forced to think about their decision during the time it takes for the alarm to be dismissed.

Another important aspect that was discussed was the ease of understanding the connection between the LED and the correct beacon. Alderick chose blue for the colour of one of the alarms (RGB: 0,0,255). This resulted in the LED blinking very similarly to the way other devices show that they are activated. Causing him to miss the link between the colour and the beacon. This shows that the unity of location is missing. There is a disconnect between the location of the LED and the location of the buttons, making that the user is confused by the function of the LED. Alderick proposed that instead of a singular led with colour, multiple LED’s at the bottom of the device should be present, showing which sections need to be activated for the alarm to turn off. In that way, the cognitive process goes from linking colour to a location to linking the sequence of the beacon to the location. This would be better because in case the user forgets what beacon to go to, they can look at the beacon shape instead of resorting to the app to check their settings.

Apart from the unity of location lacking in the current prototype, the unity of time can also be improved. Because of the inaccuracies caused by the manufacturing method of the prototype, the user is supposed to hold the device on the beacon for a second before the alarm turns off. This was done to avoid any false positives. However, this causes a delay between the user pressing the device in the beacon and the alarm turning off. Which could cause some confusion on when the device is functioning like it is supposed to. This could be solved by having a separate feedback method (like an LED) activate when the user presses the device on the beacon, no matter if this combination is correct or not.

Furthermore, the issue of the user placing the device down instead of keeping it with them became fairly apparent during the thinking-aloud process. Alderick specifically mentioned that because he is familiar with the product, he knew to keep the device in his pocket. However, normal user’s would most likely not have known this. The interaction prototype has a round-shaped top, preventing placement on that side. If this shape would be applied to both sides, the device would not have a clear orientation to be placed down. This would remove the interaction of flipping the device before putting it in the beacon, as the default position of the device is not clearly distinguishable. However, it would more strongly create the urge to put the device in your pocket as without doing so, the device would fall over.

Finally, when comparing the interaction prototype with the technical prototype, Alderick mentioned he could only think of one negative of the interaction prototype, this being the size of the beacons. The feedforward created by the beacons being able to fit in one's hand easily is that these should be placed in one's pocket, instead of at the task location. However, the size of the device itself was preferred to be small. This issue can be solved by enlarging the beacon size while still keeping the actual "coded" beacon sections the same size. An example of this can be found in figure 32.

Figure 32 - An example of how to change the beacon shape to better show its intended purpose. (hand for scale).

Some other minor remarks about the device with some added explanation:

- The sound of the alarm should be louder
    - Currently, the device uses a piezo buzzer, powered by 3.3 volts. This is pretty clear when no other sounds are present, however, when other sounds are present (e.g. when listening to music). The device can barely be heard.
- The device should have a feature that tells the user the battery almost empty
    - The device uses a rechargeable lipo battery. However, there is currently no way to tell if this battery is almost empty or not. A simple solution would be to monitor the battery level and have an LED light up when this level is below a certain threshold.
    - There is also currently no way to charge the device without taking the battery out of the device. This is not ideal for the user experience, however, this was done due to the device being a prototype. Alderick proposed the solution of having a charging dock. That way the device can be charged by simply placing the device on the dock, similar to how the device is placed on beacons.
- Alderick mentioned that the LED was very bright, which might be overstimulating to the user.
    - The LED is currently set using RGB values, for the user this is represented as a colour wheel. High RGB values correspond with very bright light. Alderick chose blue as the colour of the alarm, therefore making the RGB value 0,0,255. This resulted in a very bright blue. To prevent this from happening, a brightness slider may be integrated to give the user more control over the brightness. This can be done by letting the slider control a variable that will multiply the RGB value. For example, a less bright blue light could be achieved by having a brightness factor of 0.5. Resulting in RGB value 0,0,128.

7. Discussion

The main requirements found in section 2.4.1 were set as main goals. Requirements one and two were both achieved, as no unnecessary stimuli are present and if the stimuli currently present are overstimulating to the user they can be customised to fit their needs. Requirements 3 and 4 turned out to be less useful in the end, as no app was developed, therefore the requirements for the display of information were futile.

Apart from the main requirements, guidelines were also created to aid in designing the tangible interaction. Table 3 gives an overview of the guidelines that were achieved and the ones that were not.

Guidelines achieved	Almost achieved	Guidelines not achieved
Digital elements should be mapped to physical ones, and the other way around [22]	Unity of time	Input should be space-multiplexed [22]
Intangible aspects should be used to complement tangible ones [22]	Unity of location	Unity of expression
Interactions with tangible aspects should be based on common knowledge [22]		Unity of direction
There should be a clear continuity between tangible and intangible parts (isomorph effect) [21], [22], [25]		Unity of dynamics

Lightweight interaction should accompany the main interaction [22]		
Tangible components should be relevant [22]		
Tangible components should be expressive [22]		
Unity of modality		

Table 3 - Overview of guidelines achieved and not achieved

One of the reasons the new Ritme is lacking in regards to some guidelines is because quite a few design constraints were present. Firstly, the design needed to be simple to use and should be operated by one person, making the design have space-multiplexed inputs would negate both of these. Space-multiplexing concerns the ability of multiple users to use the product simultaneously, for example, the sandscape in figure 11. Ritme is specifically designed to be used by one person. Therefore incorporating space-multiplexing is pointless.

For other guidelines, in some cases it is simply impossible or impractical to implement all types of unity, for example, taking into account expressiveness by measuring the force the user uses to turn off the alarm can be seen as being excessive. Therefore, for good reason, Wensveen et al. state: “full unification on all the aspects may be difficult or even undesirable to achieve because intuitive interaction needs to be balanced with technology, ergonomics, production costs or aesthetics” [26, p. 3]. For that reason, I do believe a good balance has been struck between what types of unity were integrated and what types were left out. Both implementing unity of expression and unity of dynamics would have, in my opinion, diminished the user experience, as interaction with the device would become obscure. Furthermore, the digital act of an alarm being turned off does not have a direction, therefore, implementing unity of direction is not possible. One could argue that the alarm being turned off is similar to turning the volume down on the device, making down the direction of interaction. In that case, the new device already includes the unity of direction, as you push the device down to deactivate the alarm. However, this connection is a bit far-fetched.

The missing unities of location and time were intended to be in the prototype, however, during the evaluation, these unities were found to be lacking. These can both be easily fixed as discussed before, it is important to do so as the user experience should not be confusing.

The results from the second user test are also interesting to consider. The outcome was that instead of preferring a truly representative action, users would rather have a simple interaction method with which they are already familiar. There is a clear difference between user familiarity and interaction resemblance. The theory on tangible interaction stated that it is very important for the actions of the user to reflect the action performed digitally. Furthermore, there should be a clear continuity between tangible and intangible parts. And if this was not possible, further instructions should be given. In contrast, this user test clearly showed that simpler, more familiar actions were preferred over options that had a stronger resemblance to the digital action.

8. Conclusion

The goal of this project was to create a functional prototype that could take over some of the functionalities of the Ritme app while also incorporating tangible interaction to improve user experience. For this two prototypes were created. One focused on the technical aspects and the other focused on the interaction between the user and the product. The technical prototype proved that the concept was feasible in terms of electronics. The interaction prototype showed that the created concept is still lacking in some features as the unity of time and unity of location were not implemented well. These can however be easily improved by implementing some minor changes. In terms of tangible interaction, it was found that not all guidelines found were applicable in this case. Therefore, future designers that want to integrate tangible interaction should choose which guidelines are most applicable to their project, order these on importance and act accordingly. Furthermore, designers may want to focus more on the user's familiarity with the interaction as opposed to the classical tangible interaction way of focusing on the link between digital and physical actions, this is however based on the results of a small user test.

During this project, the creative technology design process was used. Consisting of diverging and converging phases. This proved useful as the constant iteration ended up improving the outcome significantly. Within this process, a similar brainstorming technique to object brainstorming was used. This further proved useful as it made it possible to quickly generate ideas without the necessity of writing out every interaction.

Overall, taking into account Alderick's opinions about the prototypes, and the fact that all applicable guidelines were implemented or can be implemented with very easy changes, I would say a successful prototype has been created.

9. Future work

During this graduation project, the main focus was on designing the components of Ritme outside of the app. This was done because the Ritme app already exists, therefore, adjusting that during the prototyping phase is very time-consuming. The technical prototype currently works via a Bluetooth serial app [40]. Therefore, the integration of the Bluetooth functionality into the Ritme app should be further developed as right now this has not been looked at. Furthermore, during the evaluation with Alderick, it became clear that the app should firstly not allow the user to set up a beacon without any raised section (as this doesn't exist), and that the app should have a brightness slider along with the colour picker for an easier user experience.

As discussed in section 2.4.3, during the state of the art research, other ideas were found that may be useful in future developments for the Ritme. These ideas were not included in the current design cycle as they all relate to the development of the app. A quick overview of these ideas is given here again:

1. Community support can be a great motivator [32]
2. Having quick access to, or allowing the caretaker to see progress is appreciated. [33], [35]
3. Relaxation and reflection methods can be included [34], [35]
4. Allowing users to get an overview of their tasks is appreciated [30]–[34]
5. Positive reinforcement can be used to stimulate continued use [21]
6. Markers can be used to show the progress of the user [21]

Apart from app features, the current prototype can also be refined. The current prototype was tested with Alderick, however, the opinions and experiences of real end-users have not been evaluated. The evaluation protocol that was used for Alderick can in theory also be used for people less familiar with the inner workings of the Ritme. Including more people in the evaluation phase would benefit the device massively. By gathering user feedback the device can be improved based on that feedback instead of on the creator's intuition, resulting in pro-actively avoiding problems experienced by the users.

One of the main things that should be changed in future prototypes is the number of functionalities that the device takes over from the app. This to increase phone battery life and to have more accurate sensors that the app can use, as this was mentioned by people currently using the Ritme app. The current prototype is focused on saving the battery life of the physical device and making the phone do the heavy lifting, therefore, changing the focus to saving the battery life of the phone requires an entirely new way of thinking and an entirely new prototype.

As discussed before in section 6, during the evaluation with Alderick other potential improvements were found. These are:

- A dismiss alarm button that needs to be held to activate.
- An LED per beacon section at the bottom of the device to indicate which sections are needed to deactivate the alarm
- A rounded form on both sides of the device, to discourage putting the device down
- A charging dock
- A brightness slider in the colour picker section of the app

Apart from working on the Ritme future research could also focus on other aspects. As discussed before in section 2.1, currently the focus of designing for neurodiverse people is on overcoming their weaknesses. While this is very noble, it paints the picture that there are no upsides to being neurodiverse, even though this is not the case. Future research can be performed to put a stronger focus on designing for the strengths of neurodiverse people instead of avoiding their weaknesses.

Furthermore, the results of the second user tests can be investigated further. It would be interesting to see whether users prefer an interaction with which they are already familiar over a tangible interaction with less focus on this familiarity. The results of the second user test seem to indicate that interaction familiarity is more important. However, this was the result of a small scale online user test. For future research different types of interactions can be created and have a large group of users physically perform these interactions.

Finally, during this research, no focus was put on the eventual manufacturing method of the device. 3D printing was used as it is very flexible and easily accessible by me (since I have a 3D printer myself). However, this method of manufacturing is not viable for mass production as it is quite expensive, prone to errors and very time-consuming. Methods like injection moulding or thermoforming might prove more useful. These methods also provide better tolerances, therefore, making designing for a specific fit easier. The best manufacturing method should be found based on variables like cost of startup, expected sales, and more. After this, the design of the device should be adjusted to better fit the chosen manufacturing method. Along with this, the size of the electronics should be reduced in size. A microcontroller was used for the prototype, which is great for prototyping, however, it is quite substantial. A custom printed circuit board could be created that only includes the features necessary for the device to function. This would greatly reduce the size required for the device.

References

[1] Harvey Blume, ‘Neurodiversity - Harvey Blume - The Atlantic’, 1998. [Online]. Available: https://archive.is/20130105003900/http://www.theatlantic.com/doc/199809u/neurodiversity

[2] S. Michie *et al.*, ‘The behavior change technique taxonomy (v1) of 93 hierarchically clustered techniques: Building an international consensus for the reporting of behavior change interventions’, *Ann. Behav. Med.*, vol. 46, no. 1, pp. 81–95, Aug. 2013, doi: 10.1007/s12160-013-9486-6.

[3] E. L. Deci and R. M. Ryan, ‘Self-determination theory.’, in *Handbook of theories of social psychology, Vol. 1*, Thousand Oaks, CA: Sage Publications Ltd, 2012, pp. 416–436. doi: 10.4135/9781446249215.n21.

[4] E. Luria, M. Shalom, and D. A. Levy, ‘Cognitive Neuroscience Perspectives on Motivation and Learning: Revisiting Self-Determination Theory’, *Mind Brain Educ.*, vol. 15, no. 1, pp. 5–17, Feb. 2021, doi: 10.1111/mbe.12275.

[5] A. T. Jersild, *Mental set and shift*. New York, 1927. [Online]. Available: https://archive.org/details/mentalsetshift00jers/page/n3/mode/2up

[6] K. K. Wu, V. Anderson, and U. Castiello, ‘Developmental Neuropsychology Neuropsychological Evaluation of Deficits in Executive Functioning for ADHD Children With or Without Learning Disabilities’, 2010, doi: 10.1207/S15326942DN2202_5.

[7] E. L. Hill, ‘Executive dysfunction in autism’, *Trends Cogn. Sci.*, vol. 8, no. 1, pp. 26–32, 2004, doi: 10.1016/j.tics.2003.11.003.

[8] J. van Dijk, M. Kopke, N. van Huizen, L. van Uffelen, and L. Beunk, ‘Empowering young adults on the autistic spectrum: Reframing assistive technology through design’, Apr. 2019, doi: 10.6084/m9.figshare.7855907.v2.

[9] N. Tromp, P. Hekkert, and P.-P. Verbeek, ‘Design for Socially Responsible Behavior: A Classification of Influence Based on Intended User Experience’, *Des. Issues*, vol. 27, pp. 3–19, Jul. 2011, doi: 10.1162/DESI_a_00087.

[10] É. Duke and C. Montag, ‘Smartphone addiction, daily interruptions and self-reported productivity’, *Addict. Behav. Rep.*, vol. 6, pp. 90–95, Dec. 2017, doi: 10.1016/j.abrep.2017.07.002.

[11] J. Dora, M. van Hooff, S. Geurts, M. Kompier, and E. Bijleveld, ‘Labor/leisure decisions in their natural context: The case of the smartphone’, *Psychon. Bull. Rev.*, pp. 1–10, Nov. 2020, doi: 10.3758/s13423-020-01844-2.

[12] T. Djajadiningrat, B. Matthews, and M. Stienstra, ‘Easy doesn’t do it: Skill and expression in tangible aesthetics’, in *Personal and Ubiquitous Computing*, Dec. 2007, vol. 11, no. 8, pp. 657–676. doi: 10.1007/s00779-006-0137-9.

[13] C. Frauenberger, J. Good, A. Alcorn, and H. Pain, ‘Conversing through and about technologies: Design critique as an opportunity to engage children with autism and broaden research(er) perspectives’, *Int. J. Child-Comput. Interact.*, vol. 1, no. 2, pp. 38–49, May 2013, doi: 10.1016/j.ijcci.2013.02.001.

[14] C. Putnam and L. Chong, *Software and Technologies Designed for People with Autism: What do users want?* 2008. [Online]. Available: http://seattleaspergers.org/research.aspx.

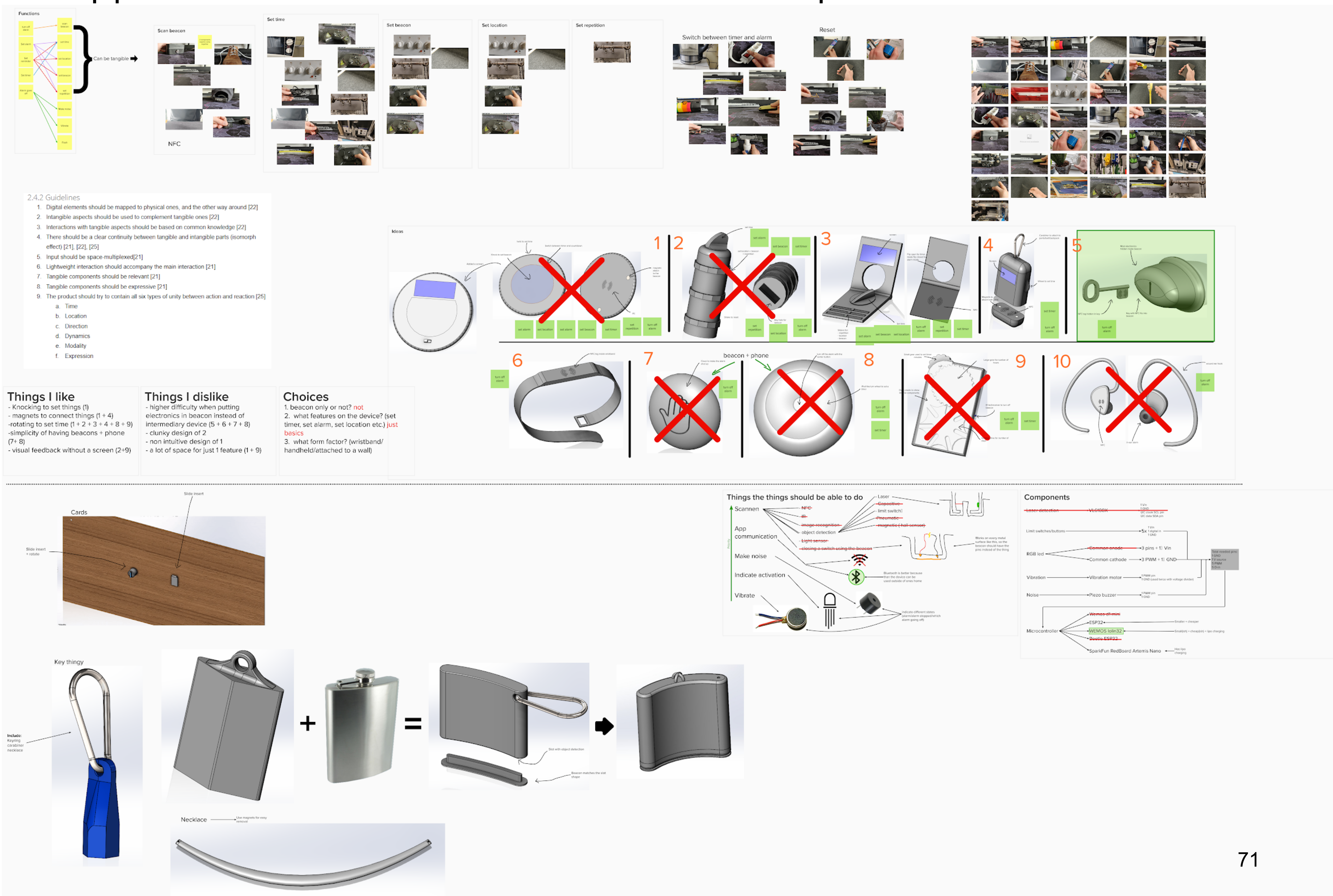
[15] C. Putnam, C. Hanschke, J. Todd, J. Gemmell, and M. Kollia, ‘2 Interactive Technologies Designed for Children with Autism: Reports of Use and Desires from Parents, Teachers, and Therapists’, *ACM Trans. Access. Comput.*, vol. 12, no. 3, pp. 12–12, 2019, doi: 10.1145/3342285.

[16] L. Bartoli, F. Garzotto, M. Gelsomini, L. Oliveto, and M. Valoriani, ‘Designing and evaluating touchless playful interaction for ASD children’, in *Proceedings of the 2014 conference on Interaction design and children*, New York, NY, USA, Jun. 2014, pp. 17–26. doi:

10.1145/2593968.2593976.
[17] S. Fletcher-Watson, ‘A Targeted Review of Computer-Assisted Learning for People with Autism Spectrum Disorder: Towards a Consistent Methodology’, 2014, doi: 10.1007/s40489-013-0003-4.
[18] S. Fletcher-Watson, H. Pain, S. Hammond, A. Humphry, and H. McConachie, ‘Designing for young children with autism spectrum disorder: A case study of an iPad app’, *Int. J. Child-Comput. Interact.*, vol. 7, pp. 1–14, Jan. 2016, doi: 10.1016/j.ijcci.2016.03.002.
[19] M. Fabri, ‘Designing digital tools for autistic students – reflections on participatory design – The IMAGE Project’, Jun. 2019. [Online]. Available: http://www.imageautism.com/designing-digital-tools-for-autistic-students-participatory-design-considerations/
[20] N. J. Rasche, ‘Design Strategy for the Development of Applications for Autism Instruction’. [Online]. Available: https://docs.lib.purdue.edu/open_access_theses://docs.lib.purdue.edu/open_access_theses/127
[21] L. McKnight, ‘Designing for ADHD: in search of guidelines’, p. 4.
[22] E. Hornecker and J. Buur, ‘Getting a grip on tangible interaction: a framework on physical space and social interaction’, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, Apr. 2006, pp. 437–446. doi: 10.1145/1124772.1124838.
[23] H. Ishii, ‘Tangible bits: beyond pixels’, in *Proceedings of the 2nd international conference on Tangible and embedded interaction*, New York, NY, USA, Feb. 2008, pp. xv–xxv. doi: 10.1145/1347390.1347392.
[24] B. van Driel, O. Mathijssen, and R.-J. Berkenbos, ‘Modsy Controller – Bringing back analog feeling to digital tools’, *Modsy*. https://modsycontroller.com/ (accessed Apr. 02, 2021).
[25] K. Overbeeke, T. Djajadiningrat, C. Hummels, S. Wensveen, and J. Prens, ‘Let’s Make Things Engaging’, in *Funology: From Usability to Enjoyment*, M. A. Blythe, K. Overbeeke, A. F. Monk, and P. C. Wright, Eds. Dordrecht: Springer Netherlands, 2004, pp. 7–17. doi: 10.1007/1-4020-2967-5_2.
[26] S. Wensveen, J. Djajadiningrat, and K. Overbeeke, ‘Interaction frogger: A design framework to couple action and function through feedback and feedforward’, Jan. 2004, vol. 177–184, pp. 177–184. doi: 10.1145/1013115.1013140.
[27] ‘SandScape’. https://tangible.media.mit.edu/project/sandscape (accessed Apr. 09, 2021).
[28] ‘Microsoft Surface Dial’, *Microsoft Store*. https://www.microsoft.com/en-us/p/surface-dial/925r551sktgn (accessed Apr. 09, 2021).
[29] D. Merrill, J. Kalanithi, and P. Maes, ‘Siftables: towards sensor network user interfaces’, Jan. 2007, pp. 75–78. doi: 10.1145/1226969.1226984.
[30] ‘Readdle’, *Readdle*. https://readdle.com/ (accessed Apr. 08, 2021).
[31] ‘Google Agenda - Download de nieuwe app voor Android en iPhone’. https://www.google.com/calendar/about/ (accessed Apr. 08, 2021).
[32] ‘Coach.me’, *Coach.me*. https://www.coach.me (accessed Apr. 08, 2021).
[33] ‘MijnEigenPlan - Eigen Regie met Structuur en Zelfredzaamheid’. https://mijneigenplan.nl/ (accessed Apr. 08, 2021).
[34] ‘Krachtplanner :: Autismekracht’. https://www.autismekracht.com/krachtplanner/ (accessed Apr. 08, 2021).
[35] ‘DayMate’, *DayMate*. https://www.daymate.nl/ (accessed Apr. 08, 2021).
[36] ‘Clocky, the original alarm clock on wheels (loud)’, *Clocky*. https://clocky.com/ (accessed Apr. 08, 2021).
[37] A. Mader and W. Eggink, ‘A design process for Creative Technology’, *Proc. 16th Int. Conf. Eng. Prod. Des. Educ. Des. Educ. Hum. Technol. Relat. E PDE 2014*, no. September, pp. 568–573, 2014.

[38] ‘Apple introduces AirTag’, *Apple Newsroom*. https://www.apple.com/newsroom/2021/04/apple-introduces-airtag/ (accessed May 04, 2021).
[39] D. Norman, ‘Affordance, conventions, and design’, *Interactions*, vol. 6, pp. 38–42, May 1999, doi: 10.1145/301153.301168.
[40] *Thingsup BLE Beacon Scanner and Logger*. Thingsup. Accessed: Jun. 14, 2021. [Online]. Available: https://https://play.google.com/store/apps/details?id=io.thingsup.blescanner&gl=NLplay.google.com/store/apps/details?id=com.macdom.ble.blescanner&hl=en&gl=US

Appendix 1 - Mural as result of the ideation phase



Appendix 2 - Online questionnaire 1

Ritme user feedback

Lees alstublieft eerst de informatiebrochure en vul de consent form in voordat u deze enquête invult!

The name and photo associated with your Google account will be recorded when you upload files and submit this form

Not **r.n.venhuizen@student.utwente.nl**? Switch account

* Required

Als u de informatie brochure heeft gelezen en de consent form heeft ingevuld upload dan het ondertekende consent form hier. *

Add file

Functies

De Ritme app heeft nu een paar functies
1. Een QR code scannen om het alarm uit te zetten
2. Geluid maken als het alarm afgaat
3. Tijd instellen wanneer een alarm af gaat (bijvoorbeeld 14:30)
4. Tijd instellen van een timer (bijvoorbeeld over 30 minuten)
5. Voor een alarm kiezen bij welke QR code het hoort
6. Voor een alarm instellen of hij herhaalt moet worden (bijvoorbeeld elke week)
7. Voor een alarm instellen op welke plek hij af moet gaan (bijvoorbeeld alleen afgaan als u thuis bent)

Welke van deze functies zou u liever op een los apparaat doen in plaats van op uw telefoon? En waarom? Let op elke extra functie op een fysiek apparaat maakt het duurder om te kopen. *

Long answer text

Feedback

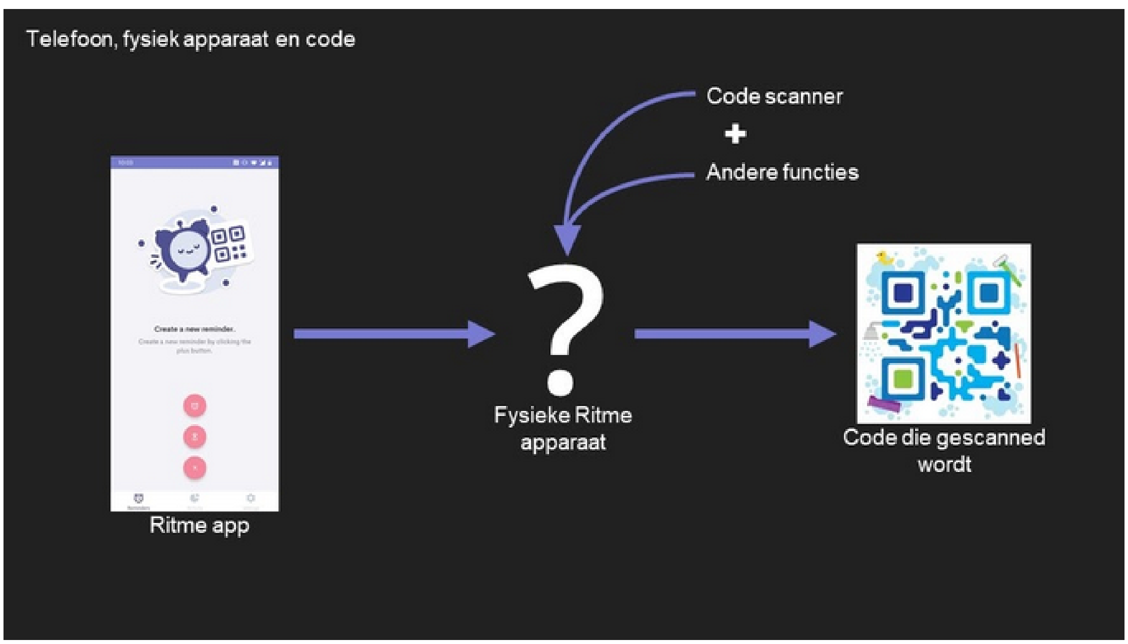
Description (optional)

Op welke manier(en) zou u willen dat de Ritme u op de hoogte stelt *

- [ ] Een geluid (zoals een wekker)
- [ ] Trillingen (zoals een telefoon)
- [ ] Lichtflitsen
- [ ] Other...

Voor de volgende vraag is het belangrijk om te begrijpen wat er bedoeld wordt, daarom nu eerst afbeeldingen en scenarios over de verschillende situaties

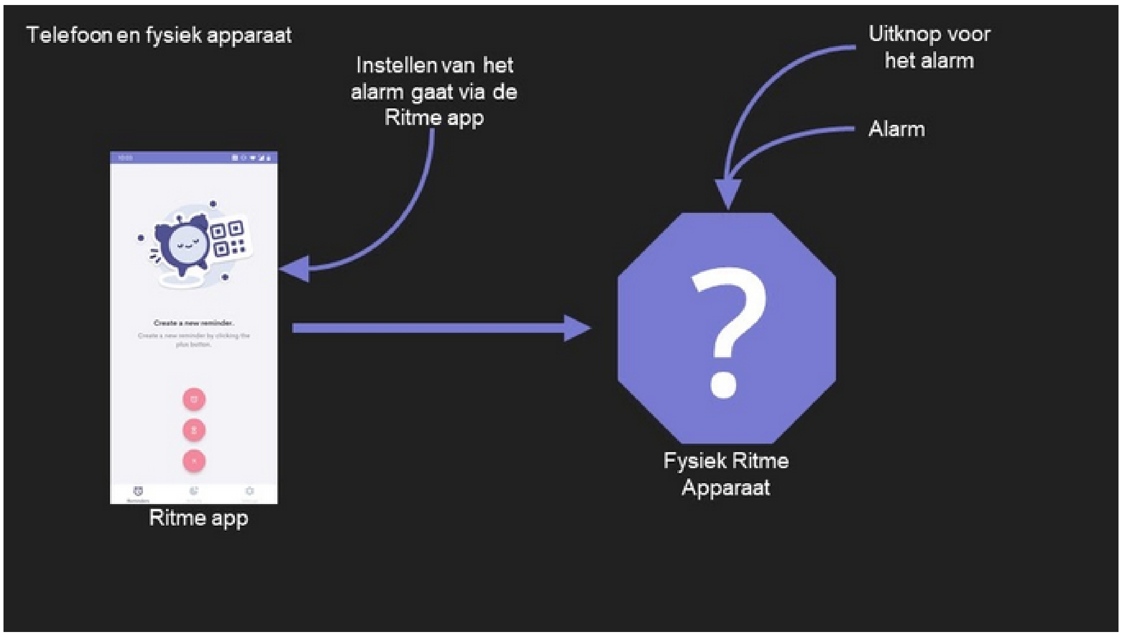
1. Een telefoon in combinatie met een los apparaat, en iets wat gescand moet worden



↑ Scenario

Frank heeft een alarm ingesteld om 7 uur 's avonds via de Ritme app zodat hij de afwas niet vergeet. Daarom heeft hij de QR code in de keuken ingesteld als locatie. Zodra het 7 uur is gaat het alarm af en legt frank zijn telefoon weg, hij pakt het fysieke Ritme apparaat, loopt daarmee naar de keuken, scant de code en begint de afwas te doen.

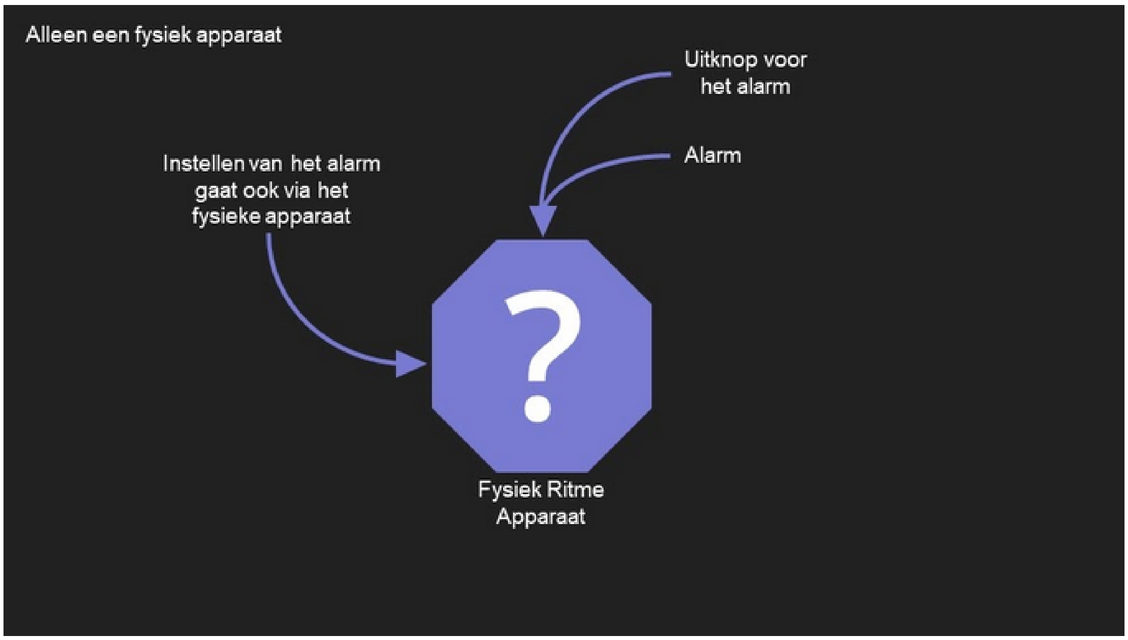
2. Een telefoon in combinatie met een los apparaat wat werkt als alarm



⭱ Scenario

Frank heeft een alarm ingesteld om 7 uur 's avonds via de Ritme app zodat hij de afwas niet vergeet. Als het 7 uur is hoort hij het alarm afgaan in de keuken. Frank legt zijn telefoon weg en loopt naar de keuken. Eenmaal in de keuken aangekomen drukt hij op de uitknop van het alarm en begint met de afwas.

3. Alleen een fysiek apparaat waarop het alarm ingesteld wordt en waarmee het alarm uitgezet kan worden



⭱ Scenario

Het is 3 uur en Frank wil om 7 uur 's avonds de afwas doen. Hij loopt daarom naar de keuken om het alarm in te stellen op het fysieke Ritme apparaat. Als het eenmaal 7 uur is loopt Frank naar de keuken toe, hij schakelt het alarm uit en begint aan de afwas. (dit idee kan gezien worden als een soort high-tech kookwekker)

Welke van de genoemde opties zou u het liefst gebruiken en waarom? *

Long answer text

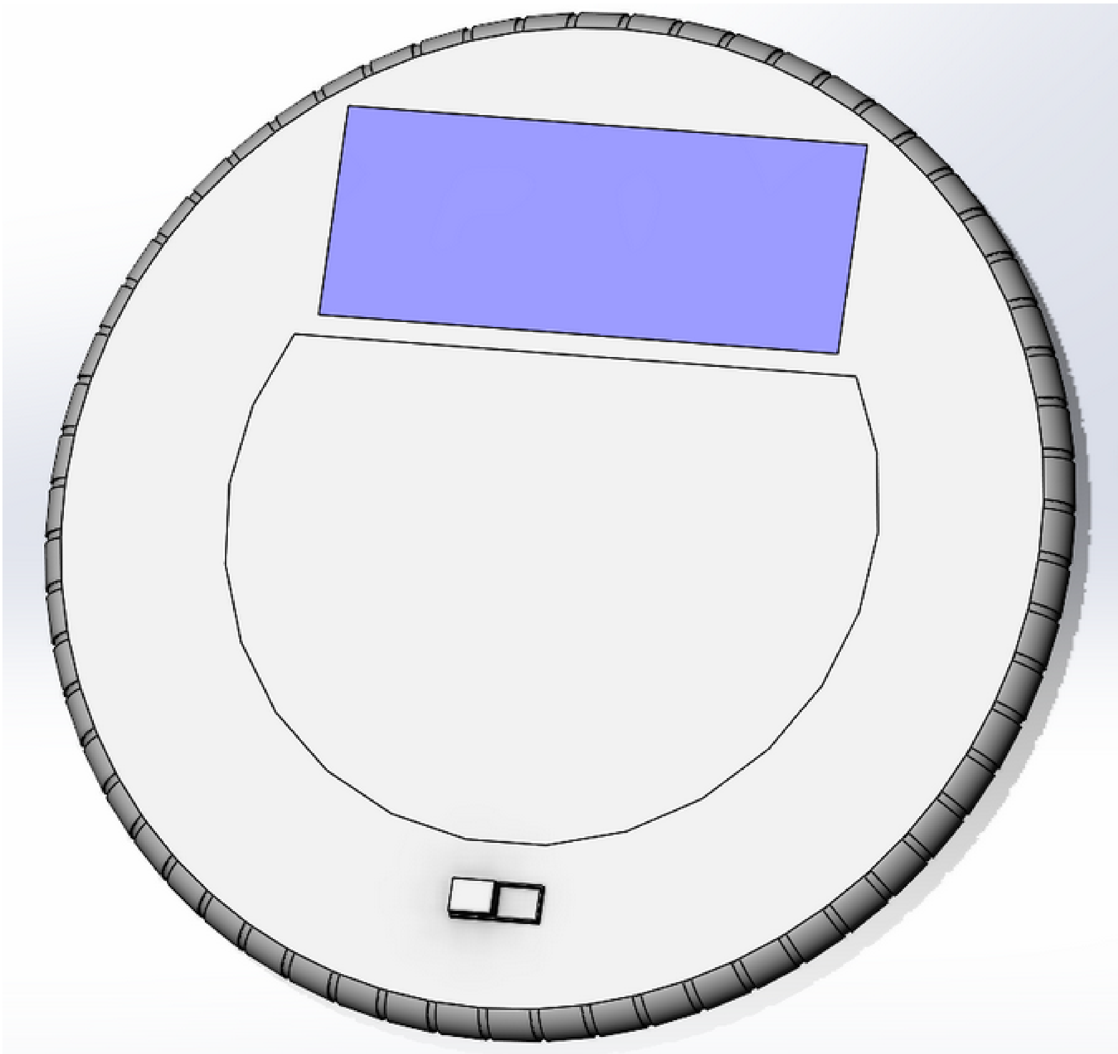
Vormgeving

In de volgende sectie, worden een paar voorbeelden gegeven van interactie. Over elk van deze voorbeelden worden een paar vragen gesteld. Het gaat hier vooral om de functies van de voorbeelden en niet zozeer het uiterlijk (al is feedback daar op ook welkom)

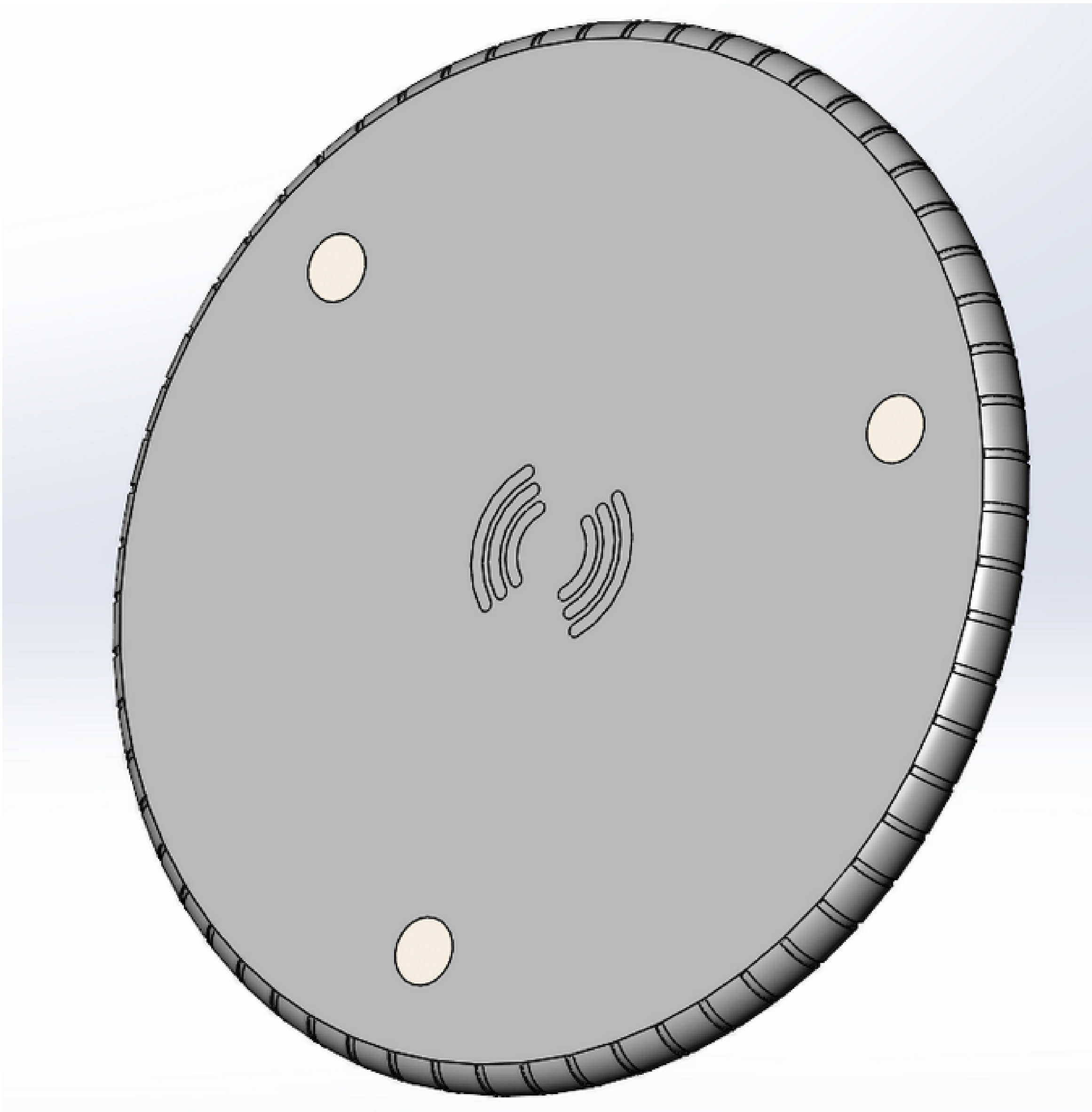
Kloppen en draaien

In dit voorbeeld ziet u een schijf die gebruikt kan worden voor alle functies die de Ritme app nu heeft. U gebruikt het door eerst een bepaald aantal keer te kloppen op het oppervlakte om te kiezen welke functie u wilt instellen, u klopt bijvoorbeeld 1 keer voor het instellen van de tijd en twee keer voor de locatie. Na het selecteren van de functie draait u aan de buitenkant om de gekozen instelling aan te passen. Om uiteindelijk het alarm uit te zetten hangt u de schijf op (op de plek waar nu de QR is) door middel van de magneetjes op de achterkant.

Voorkant



Achterkant



Wat vindt u goed aan dit voorbeeld? *

Long answer text

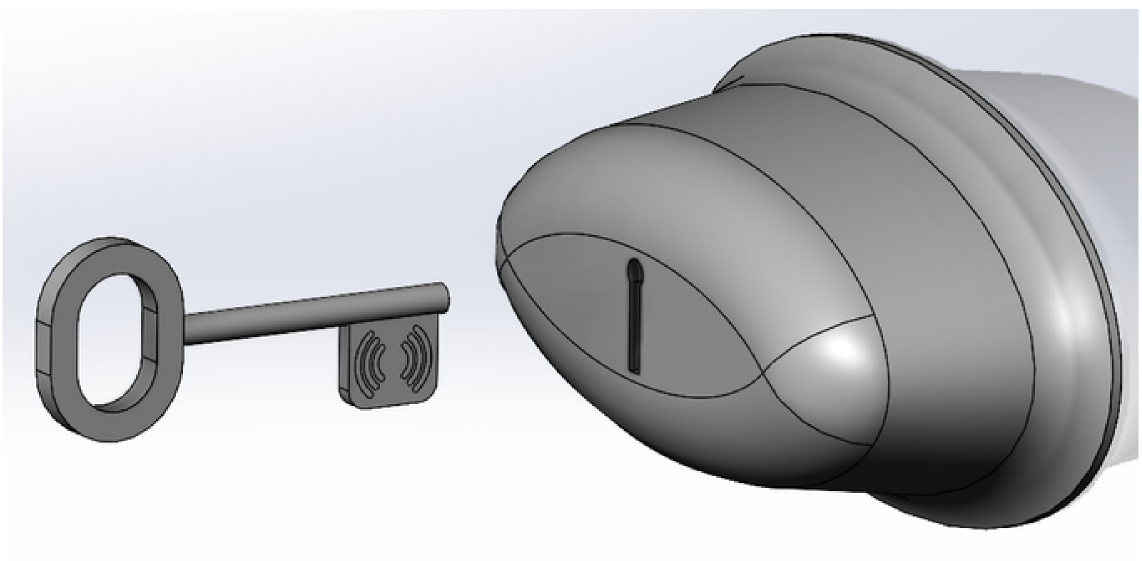
Wat vindt u slecht aan dit voorbeeld? *

Long answer text

Sleutel

In dit voorbeeld wordt in plaats van een QR code een NFC sleutelgat gebruikt. Als het alarm af gaat loopt u naar de plek van het sleutelgat en stopt u de sleutel er in. Daarna zal het alarm uit gaan. Het instellen van de alarmen wordt nog wel gedaan in de Ritme app.

Key + Key hole



Wat vindt u goed aan dit voorbeeld? *

Long answer text

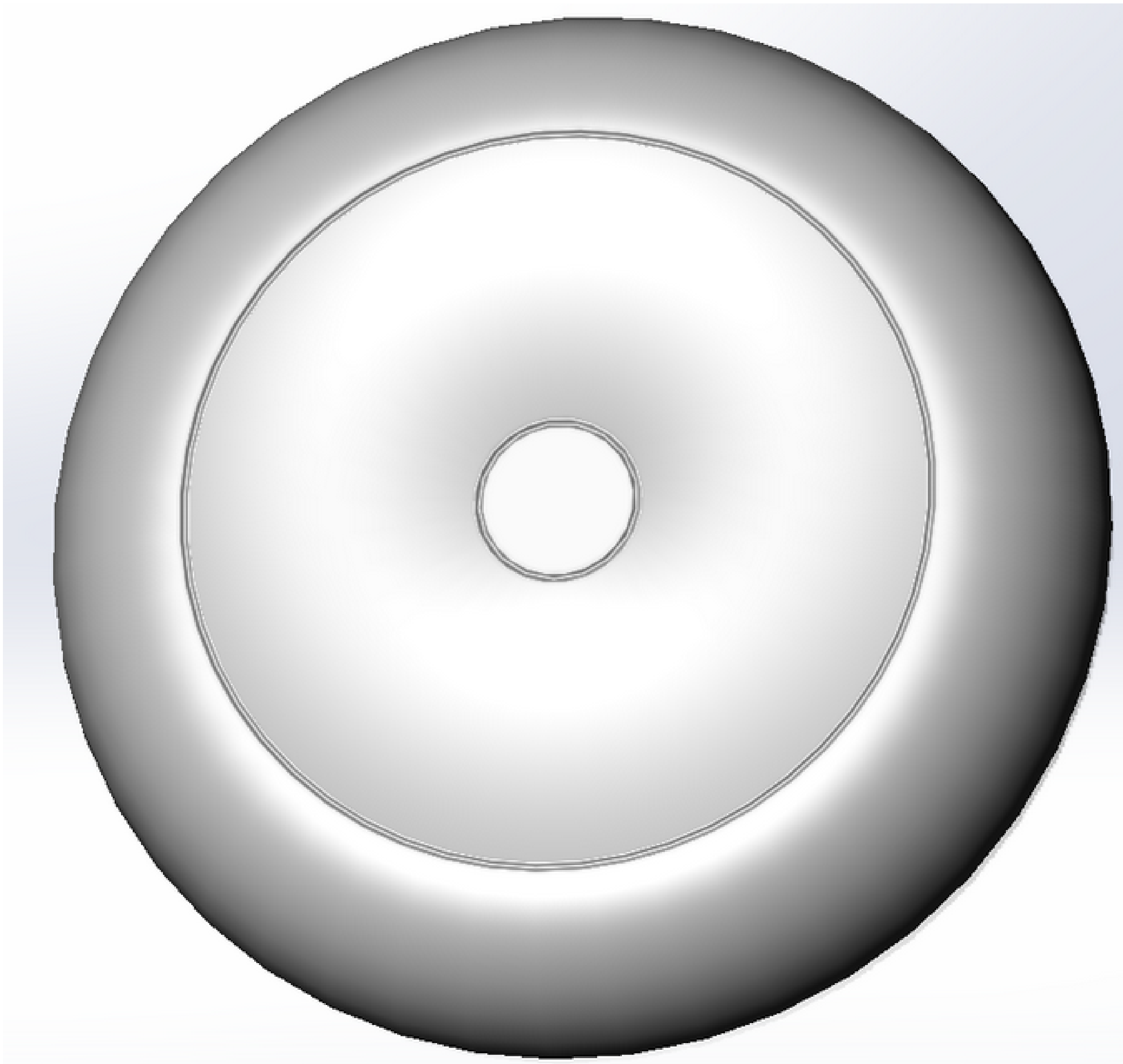
Wat vindt u slecht aan dit voorbeeld? *

Long answer text

Draaiwiel

Dit voorbeeld bestaat uit 2 onderdelen. een knop (in het midden) en een draaiwiel er om heen (vergelijkbaar met het draaiwiel van de oude IPods). Dit onderdeel vervangt de QR code. De gebruiker stelt een alarm in op zijn telefoon, als dit af gaat loopt hij/zij naar de knop en zet het uit. Verder kan de gebruiker snel een timer zetten door aan de draaiknop te draaien. Alle andere instellingen (bijvoorbeeld de tijd van het alarm) worden ingesteld via de telefoon

Front



Wat vindt u goed aan dit voorbeeld? *

Long answer text

Wat vindt u slecht aan dit voorbeeld? *

Long answer text

Klap telefoon

Dit voorbeeld kan gebruikt worden op twee manieren. Als het apparaat open is geklapt kan er een alarm worden gezet (bijvoorbeeld, morgen om 12:30). De tijd wordt dan ingesteld met de draaiknop en de andere functies zoals locatie worden ingesteld met de schuiven aan de onderkant. Als het apparaat dicht geklapt is kan de draaiknop gebruikt worden om een timer te zetten (bijvoorbeeld over 15 minuten). Om het alarm uit te zetten hou je het apparaat tegen een NFC tag opgehangen op de plek van het klusje.

Voorkant open

Wat vindt u goed aan dit voorbeeld? *

Long answer text

Wat vindt u slecht aan dit voorbeeld? *

Long answer text

Oordopje

Dit voorbeeld werkt vergelijkbaar met draadloze oordopjes.
De gebruiker koppelt dit met de Ritme app. Als het alarm af gaat hoort alleen de gebruiker dit door het oordopje heen. Hij/zij loopt vervolgens naar de plek van de taak en houdt het oordopje tegen een NFC tag aan. Daardoor zal het alarm uitgaan.

Voorkant

Achterkant

Wat vindt u goed aan dit voorbeeld? *

Long answer text

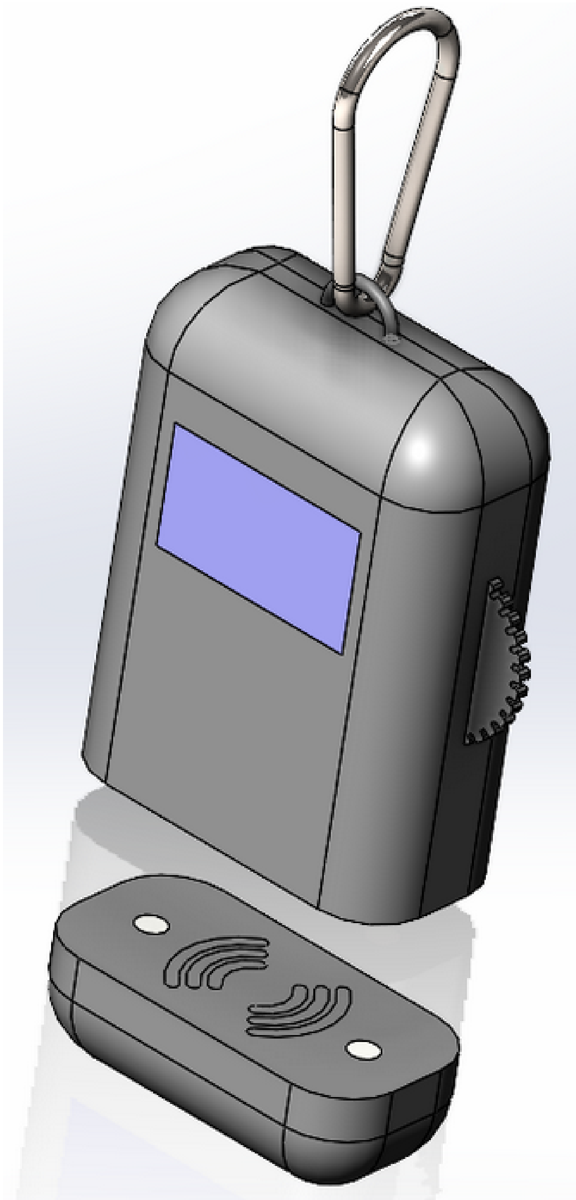
Wat vindt u slecht aan dit voorbeeld? *

Long answer text

Clip apparaat

In dit voorbeeld staat draagbaarheid centraal. Met de karabijnhaak aan de bovenkant kunt u het apparaat aan de lusjes van uw broek hangen. Het instellen van een alarm gebeurt via de Ritme app. Zodra het alarm afgaat haalt u het onderste gedeelte van het apparaat af (wat vast zit met magneetjes) en houdt het tegen de NFC tag aan op de plek van het klusje. Daarna hangt u het onderste gedeelte waar op aan de rest en kunt u uw klusje beginnen.

Voorkant



Wat vindt u goed aan dit voorbeeld? *

Long answer text

Wat vindt u slecht aan dit voorbeeld? *

Long answer text

Favoriet

Description (optional)

Welke van de voorbeelden vond u het beste? *

- het voorbeeld van de schijf met kloppen en draaien
- het sleutel voorbeeld
- Het draaiwiel voorbeeld
- Klap telefoon
- Het oordopje voorbeeld
- Clip apparaat

Waarom heeft u voor deze gekozen?

Long answer text

Onduidelijkheden

Description (optional)

Was er nog iets onduidelijk tijdens het invullen van deze enquête? Zo ja, wat?

Long answer text

Einde

Heel erg bedankt voor het invullen van deze enquête! Klik op submit als u tevreden bent met uw antwoord.

Appendix 3 - Online questionnaire 2

RITME

Ritme interaction design

Information brochure ritme

Thanks for opening this form. This form will be related to the Ritme app. Ritme is an app developed to help neurodiverse people overcome the challenge of having a successful everyday routine. This is done by using QR codes. Users will set an alarm for a certain task at a certain time. When that time comes and the users want to turn off the alarm they are required to scan the QR code at the task location. More information can be found on the Ritme website: https://ritme.codes/.

The goal of the research is to improve the interaction with the Ritme by introducing a physical component. Therefore, to find the best interaction possible we would like your help. In this questionnaire, you will be asked to answer a question related to interaction as part of the new Ritme concept. The answers you give to these questions will be stored safely, apart from this, no further information of you will be recorded or stored. As this research is done for Ritme they will also have access to the results of this questionnaire if asked. You are free to withdraw from the research at any time, no questions asked. Before filling in the question, please read and fill in the consent form on the next page.

Sincerely,
Robin Venhuizen

r.n.venhuizen@student.utwente.nl

Supervisor: Jelle van Dijk

jelle.vandijk@utwente.nl

Faculty of EEMCS, University of Twente

Ethics Committee Computer & Information Science:

Phone number: +31534892085

E-mail: ethicscommittee-cis@utwente.nl

Building: Zilverling 1051

Next

Consent form

I hereby declare that I have been informed in a manner which is clear to me about the nature and method of the research as described in the aforementioned information brochure 'Information brochure Ritme'. My questions have been answered to my satisfaction. I agree with my own free will to participate in this research. I reserve the right to withdraw this consent without the need to give any reason and I am aware that I may withdraw from the experiment at any time. If my research results are to be used in scientific publications or made public in any other manner, then they will be made completely anonymous. My personal data will not be disclosed to third parties without my express permission. If I request further information about the research, now or in the future, I may contact Robin Venhuizen. If you have any complaints about this research, please direct them to the secretary of the Ethics Committee of the Faculty of Computer & Information Science at the University of Twente, P.O. Box 217, 7500 AE Enschede (NL), email: ethicscommittee-cis@utwente.nl).

Signature *

Sign Here

Clear

Name *

Before filling in the question please use this website to spin the wheel:

https://wheelofnames.com/exy-69b

What result did you get from the wheel? *

- ◯ Questionnaire 1
- ◯ Questionnaire 2

The new ritme concept works in the following way:
1. you set an alarm for the time and place of a task (for example 20:00 kitchen)

2. the alarm goes off at the time

3. you take the physical device to the task location

4. you use the physical device to turn off the alarm using the "beacon" at the task location (originally the QR-code).

5. you do the task

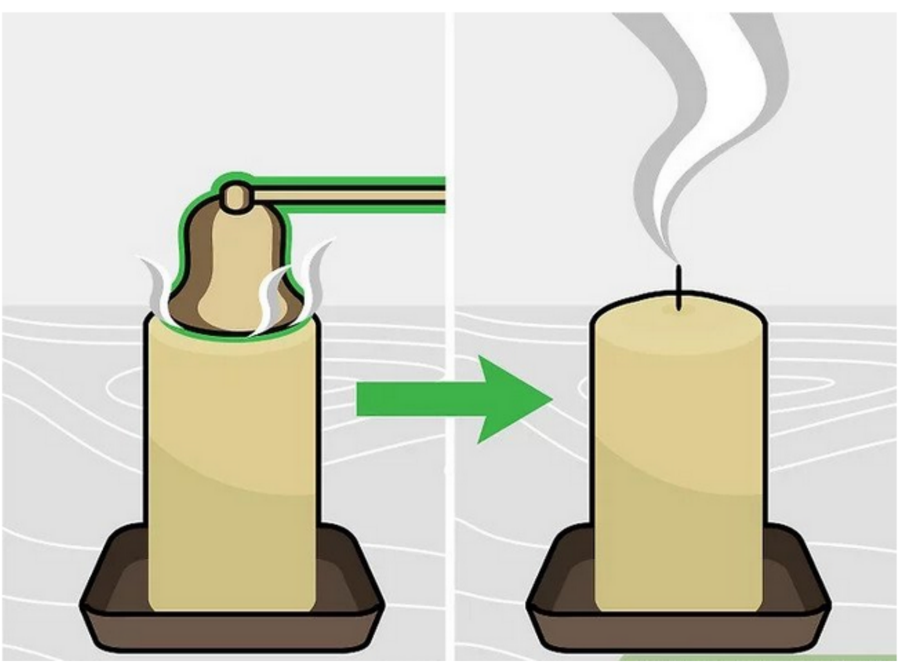
6. you use the physical device to notify the app that you have successfully completed the task.

In step 4, the way in which the alarm is turned off is not clear yet. Five different options for this are presented

1. put the alarm inside of the beacon. Similar to the childrens game below.

2. slide the alarm through the beacon. similar to swiping with a pin-code reader/pay terminal

3. putting the alarm over something else. Similar to putting a candle snuffer over a candle.



4. holding the alarm to the beacon. similar to paying with your phone.

5.hanging the alarm onto the beacon. Similar to a coat rack.

Which of these interactions do you think fit best with the task of turning off the Ritme's alarm?

ex: 23

Fill in a number from 1 to 5

Why did you choose this interaction?

Type here...

The new ritme concept works in the following way:
1. you set an alarm for the time and place of a task (for example 20:00 kitchen)

2. the alarm goes off at the time

3. you take the physical device to the task location

4. you use the physical device to turn off the alarm using the "beacon"   at the task locatic (originally the QR-code).

5. you do the task

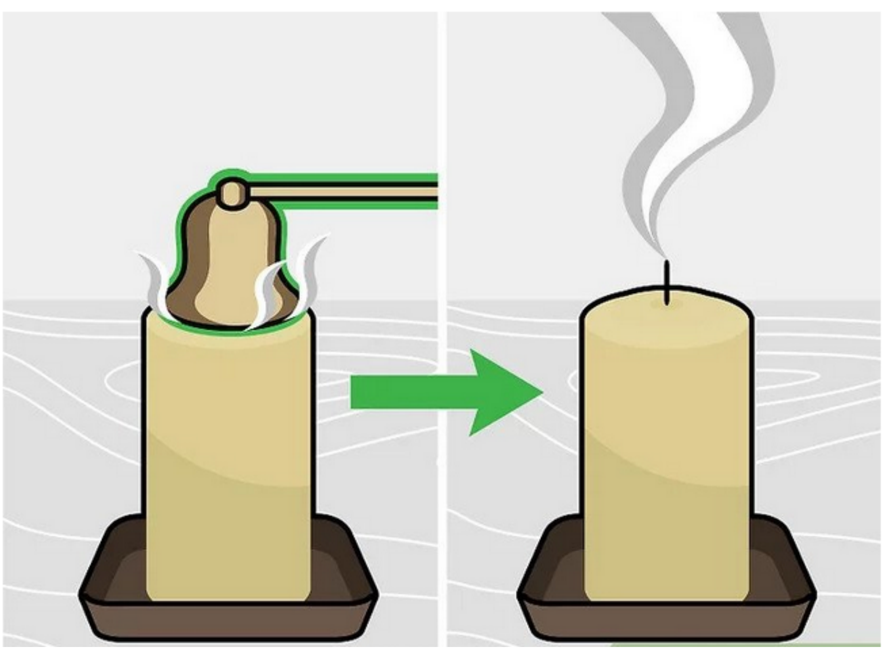
6. you use the physical device to notify the app that you have successfully completed the task.

In step 6, the way in which Ritme is notified of a successfull task is not clear yet. Five different options for this are presented

1. Take the alarm out of something. similar to removing a plug from the wall.

2. slide the physical device through the beacon. similar to swiping with a pin-code reader/pay terminal

3. taking the physical device off of something else. Similar to picking up a glass from a coaster.



4. holding the physical device to the beacon. Similar to paying with your phone.

5.Taking the physical device from the beacon. Similar to a grabbing a coat from a coat rack.

Which of these interactions do you think fit best with notifying the Ritme you are done with your task?

ex: 23

Fill in a number from 1 to 5

Why did you choose this interaction?

Type here...

Appendix 4 - Arduino code

```
//Created for Creative Technology Graduation Project
//University of Twente
//As requested by Ritme (https://ritme.codes/)
//Created by Robin Venhuizen (s2114186)
//Finalised on 15-06-2021

volatile bool button1Pressed = false;
volatile bool button2Pressed = false;
volatile bool button3Pressed = false;
volatile bool button4Pressed = false;
volatile bool button5Pressed = false;
const int buttonHold = 500;
const int debounceTime = 50;

const int buttonPin1 = 15;
const int buttonPin2 = 2;
const int buttonPin3 = 4;
const int buttonPin4 = 16;
const int buttonPin5 = 17;

const int buzzerPin = 13;
const int vibrationPin = 19;
const int rLed = 25;
const int gLed = 26;
const int bLed = 27;
bool blinkState = false;
volatile int red = 0;
volatile int green = 0;
volatile int blue = 0;

volatile int alarm_1 = 0;
volatile int alarm_2 = 0;
volatile int alarm_3 = 0;
volatile int alarm_4 = 0;
volatile int alarm_5 = 0;
volatile int soundChoice = 0;

bool resetCall = false;
volatile bool alarm_on = false;
```

```
int ticks = 0;


//------------------------------------------------ ↓ Bluetooth ↓ ----------------------------------------------- //
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEServer.h>

#define SERVICE_UUID         "d2fb211b-b520-4513-886e-8a8e021ded36"
#define CHARACTERISTIC_UUID "8f8e8b25-457f-47c3-a66c-e12eefc4b370"

char valueArray[30];

class MyCallbacks: public BLECharacteristicCallbacks
{
	void onWrite(BLECharacteristic *pCharacteristic)
	{
	std::string value = pCharacteristic->getValue();
	char * pch;

	if (value.length() > 0)
	{
	alarm_on = true; //if input is received, turn the alarm on
	for (int i = 0; i < value.length(); i++)
	{
	valueArray[i] = value[i];
	}

	if (strcasecmp(valueArray, "reset") == 0) {
	resetCall = true;
	} else {
	char *token;
	token = strtok(valueArray, ","); //split the incomming signal into chunks seperated by a comma
	int i = 0; //use variable i as a way to track which section of the input is being read.
	while ( token != NULL ) {
		printf( " %s\n", token );
		if (i == 0) { //for each section of the beacon, read what is needed to turn the alarm off
		alarm_1 = atoi(token);
		}
		if (i == 1) {
		alarm_2 = atoi(token);
		}
		if (i == 2) {
		alarm_3 = atoi(token);
```

```
            }
            if (i == 3) {
            alarm_4 = atoi(token);
            }
            if (i == 4) {
            alarm_5 = atoi(token);
            }
            if (i == 5) { //Read the Red value
            red = atoi(token);
            }
            if (i == 6) { //Read the Green value
            green = atoi(token);
            }
            if (i == 7) { //Read the Blue value
            blue = atoi(token);
            }
            if (i == 8) {
            soundChoice = atoi(token);
            }

            i++;
            token = strtok(NULL, ",");
        }
        i = 0;
        }
        memset(valueArray, 0, sizeof(valueArray)); //clear the array after checking
        }
        }
};
//------------------------------------------------- ↑ Bluetooth ↑ ------------------------------------------------- //

//timers for blink duration
unsigned long previousMillisBlink = 0;
unsigned long currentMillisBlink = 0;

//------------------------------------------------- ↓ Different sounds ↓ ------------------------------------------ //
unsigned long previousMillis = 0;
boolean outputTone = false;                // Records current state
int MelodyIndex = 0;
unsigned long currentMillis;

//standard melody, maximum number of different notes is 7
const int MelodyLength = 7;
int Melody[MelodyLength] = {660, 660, 660, 510, 660, 770, 380};
```

```
int pauseLength[MelodyLength] = {475, 50, 200, 200, 0, 200, 450};
int noteLength[MelodyLength] = {100, 100, 100, 100, 100, 100, 100};

//melodies that van overwrite the standard one
int Melody1[MelodyLength] = {660, 660, 660, 510, 660, 770, 380};
int pauseLength1[MelodyLength] = {475, 50, 200, 200, 0, 200, 450}; //everything is shifted one to the right
int noteLength1[MelodyLength] = {100, 100, 100, 100, 100, 100, 100};

int Melody2[MelodyLength] = {440, 660, 0, 0, 0, 0, 0};
int pauseLength2[MelodyLength] = {100, 100, 0, 0, 0, 0, 0}; //everything is shifted one to the right
int noteLength2[MelodyLength] = {500, 500, 0, 0, 0, 0, 0};


//------------------------------------------------ ↑ Different sounds ↑ ------------------------------------------ //

//Interrupts per button
void IRAM_ATTR button1Interrupt() {
  static unsigned long last_interrupt_time = 0;
  unsigned long interrupt_time = millis();
  // If interrupts come faster than 200ms, assume it's a bounce and ignore
  if (interrupt_time - last_interrupt_time > debounceTime) {
       button1Pressed = !button1Pressed;
  }
  last_interrupt_time = interrupt_time;
}

void IRAM_ATTR button2Interrupt() {
  static unsigned long last_interrupt_time = 0;
  unsigned long interrupt_time = millis();
  // If interrupts come faster than 200ms, assume it's a bounce and ignore
  if (interrupt_time - last_interrupt_time > debounceTime) {
       button2Pressed = !button2Pressed;
  }
  last_interrupt_time = interrupt_time;
}

void IRAM_ATTR button3Interrupt() {
  static unsigned long last_interrupt_time = 0;
  unsigned long interrupt_time = millis();
  // If interrupts come faster than 200ms, assume it's a bounce and ignore
  if (interrupt_time - last_interrupt_time > debounceTime) {
       button3Pressed = !button3Pressed;
  }
  last_interrupt_time = interrupt_time;
```

```
}

void IRAM_ATTR button4Interrupt() {
  static unsigned long last_interrupt_time = 0;
  unsigned long interrupt_time = millis();
  // If interrupts come faster than 200ms, assume it's a bounce and ignore
  if (interrupt_time - last_interrupt_time > debounceTime) {
       button4Pressed = !button4Pressed;
  }
  last_interrupt_time = interrupt_time;
}

void IRAM_ATTR button5Interrupt() {
  static unsigned long last_interrupt_time = 0;
  unsigned long interrupt_time = millis();
  // If interrupts come faster than 200ms, assume it's a bounce and ignore
  if (interrupt_time - last_interrupt_time > debounceTime) {
       button5Pressed = !button5Pressed;
  }
  last_interrupt_time = interrupt_time;
}

void setup() {
  Serial.begin(115200);

  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
  pinMode(buttonPin3, INPUT);
  pinMode(buttonPin4, INPUT);
  pinMode(buttonPin5, INPUT);
  pinMode(rLed, OUTPUT);
  pinMode(gLed, OUTPUT);
  pinMode(bLed, OUTPUT);
  pinMode(vibrationPin, OUTPUT);

  // configure PWM functionalities
  ledcSetup(0, 8000, 12);
  ledcSetup(1, 5000, 8);
  ledcSetup(2, 5000, 8);
  ledcSetup(3, 5000, 8);
  ledcSetup(4, 500, 8);

  // attach channels to the GPIO to be controlled
  ledcAttachPin(buzzerPin, 0);
```

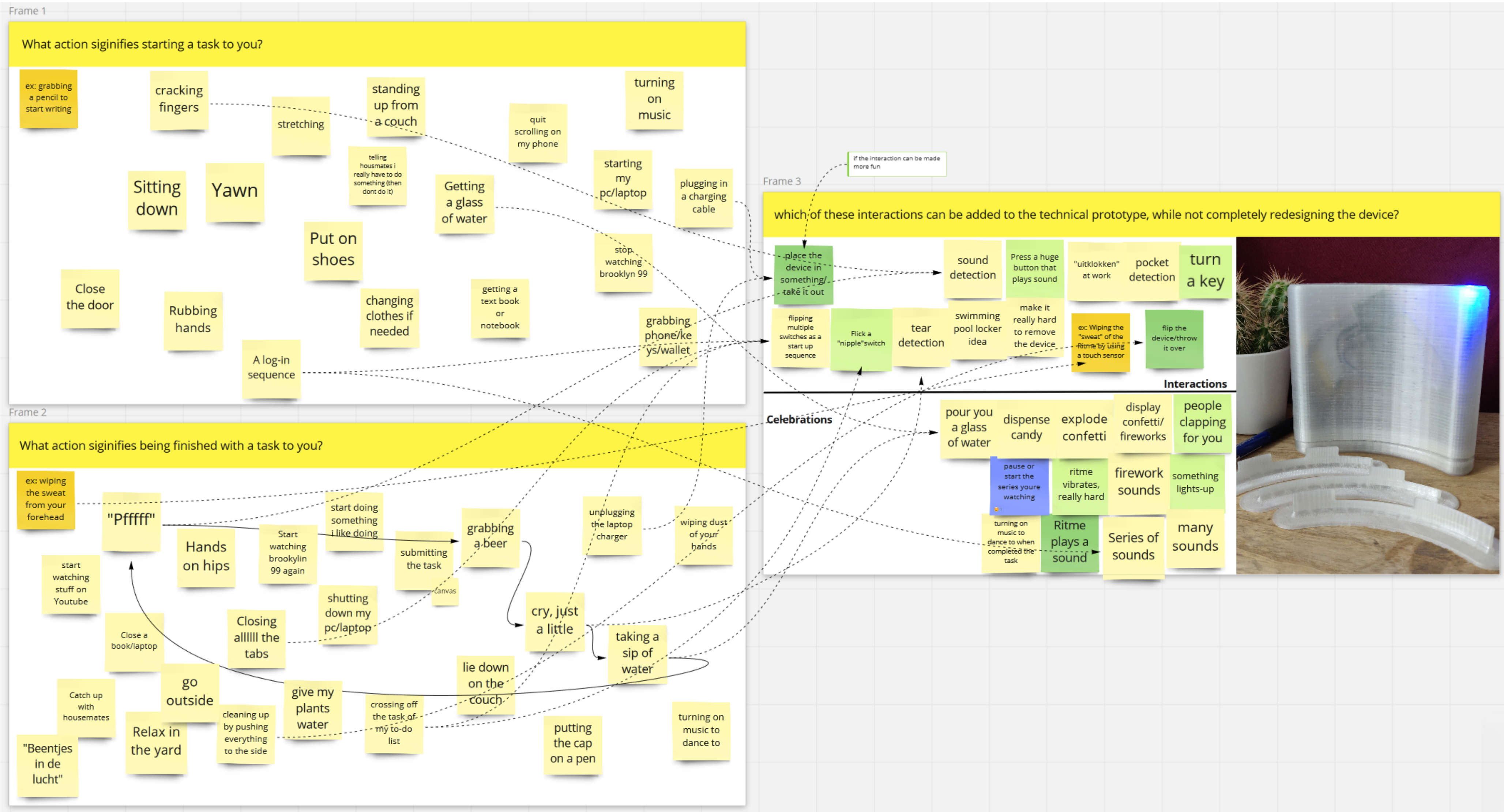
```
  ledcAttachPin(rLed, 1);
  ledcAttachPin(gLed, 2);
  ledcAttachPin(bLed, 3);
  ledcAttachPin(vibrationPin, 4);

  //Interrupt
  attachInterrupt(digitalPinToInterrupt(buttonPin1), button1Interrupt, CHANGE);
  attachInterrupt(digitalPinToInterrupt(buttonPin2), button2Interrupt, CHANGE);
  attachInterrupt(digitalPinToInterrupt(buttonPin3), button3Interrupt, CHANGE);
  attachInterrupt(digitalPinToInterrupt(buttonPin4), button4Interrupt, CHANGE);
  attachInterrupt(digitalPinToInterrupt(buttonPin5), button5Interrupt, CHANGE);

  //------------------------------------------------- ↓ Bluetooth ↓ ------------------------------------------------- //
  BLEDevice::init("Ritme+");
  BLEServer *pServer = BLEDevice::createServer();

  BLEService *pService = pServer->createService(SERVICE_UUID);

  BLECharacteristic *pCharacteristic = pService->createCharacteristic(
                                         CHARACTERISTIC_UUID,
                                         BLECharacteristic::PROPERTY_READ |
                                         BLECharacteristic::PROPERTY_WRITE
                                       );

  pCharacteristic->setCallbacks(new MyCallbacks());

  pCharacteristic->setValue("input form: x,x,x,x,x,R,G,B,x");
  pService->start();

  BLEAdvertising *pAdvertising = pServer->getAdvertising();
  pAdvertising->start();
  //------------------------------------------------- ↑ Bluetooth ↑ ------------------------------------------------- //

}

void loop() {
  currentMillis = millis();
  currentMillisBlink = millis();
  lights();
  sound();
  checkAlarm();

  if (resetCall == true) {
      reset();
```

```
  }

  //  Print to serial (useful for bug fixing)
  //  Serial.print(button1Pressed);
  //  Serial.print(alarm_1);
  //  Serial.print("\t");
  //  Serial.print(button2Pressed);
  //  Serial.print(alarm_2);
  //  Serial.print("\t");
  //  Serial.print(button3Pressed);
  //  Serial.print(alarm_3);
  //  Serial.print("\t");
  //  Serial.print(button4Pressed);
  //  Serial.print(alarm_4);
  //  Serial.print("\t");
  //  Serial.print(button5Pressed);
  //  Serial.print(alarm_5);
  //  Serial.print("\t");
  //  Serial.print("RGB: ");
  //  Serial.print(red);
  //  Serial.print(",");
  //  Serial.print(green);
  //  Serial.print(",");
  //  Serial.print(blue);
  //  Serial.print("\t");
  //  Serial.print("alarm_on:");
  //  Serial.print(alarm_on);
  //  Serial.print("\t");
  //  Serial.print("ticks: ");
  //  Serial.print(ticks);
  //  Serial.println();
}


void checkAlarm() {
  if (alarm_on == true) {
       if (button1Pressed == alarm_1 && button2Pressed == alarm_2 && button3Pressed == alarm_3 && button4Pressed == alarm_4 &&
button5Pressed == alarm_5) { //check if all button inputs are correct
       ticks++;
       if (ticks > buttonHold) { //check if correct buttons are pressed for longer than the set buttonHold
       //     turn off alarm
       resetCall = true;
       ticks = 0;
       }
```

```
        delay(1);
        } else {
        ticks = 0; //reset button hold counter
        }
  }
}


void lights() {
  if (alarm_on == true) {
        blinkLed(red, green, blue, 500);
  }
}

void blinkLed(int r, int g, int b, int blinkSpeed) {
  if (currentMillisBlink - previousMillisBlink >= blinkSpeed) {
        previousMillisBlink = currentMillisBlink;
        if (blinkState == true) {
        ledcWrite(1, 0);
        ledcWrite(2, 0);
        ledcWrite(3, 0);
        blinkState = false;
        } else {
        ledcWrite(1, r);
        ledcWrite(2, g);
        ledcWrite(3, b);
        blinkState = true;
        }
  }
}

void sound() {
  if (alarm_on == true) {
        if (soundChoice == 0) { //check which melody was chosen
        for (int i = 0; i <= MelodyLength; i++) {
        Melody[i] = Melody1[i];
        pauseLength[i] = pauseLength1[i];
        noteLength[i] = noteLength1[i];
        }
        } else if (soundChoice == 1) { //check which melody was chosen
        for (int i = 0; i <= MelodyLength; i++) {
        Melody[i] = Melody2[i];
        pauseLength[i] = pauseLength2[i];
        noteLength[i] = noteLength2[i];
```

```
      }
      }

      if (outputTone) {
      // We are currently outputting a tone
      // Check if it's been long enough and turn off if so
      if (currentMillis - previousMillis >= noteLength[MelodyIndex]) {
      previousMillis = currentMillis;
      ledcWriteTone(0, 0);
      ledcWrite(4, 0); //vibration motor
      outputTone = false;
      }
      } else {
      // We are currently in a pause
      // Check if it's been long enough and turn on if so
      if (currentMillis - previousMillis >= pauseLength[MelodyIndex]) {
      previousMillis = currentMillis;
      ledcWriteTone(0, Melody[MelodyIndex]);
      ledcWrite(4, 255); //vibration motor
      outputTone = true;
      //Update to play the next tone, next time
      MelodyIndex = MelodyIndex + 1;
      if (MelodyIndex >= MelodyLength) {
      MelodyIndex = 0;
      }
      }
      }
  }
}

void reset() { //turn off all alarm methods
  ledcWriteTone(0, 0);
  ledcWrite(1, 0);
  ledcWrite(2, 0);
  ledcWrite(3, 0);
  ledcWrite(4, 0);
  MelodyIndex = 0;
  alarm_on = false;
  resetCall = false;
}
```

Appendix 5 - Interaction brainstorm session



Appendix 6 - Evaluation protocol for final usability test

Hypothesis:
"The user can independently figure out how to turn off the alarm, through the perceived affordances created by the product"
"The user can turn off the alarm without referring to their phone"
"The user can identify which location to go to through the combinations of colours and melodies"

Introduce technical prototype

Setup
Imagine you're setting up the ritme app. On this piece of paper, all the (new) options for the alarms are presented. Please set two alarms and move the beacons to the correct location

Alarm 1 -> 10 uur in de keuken -> beep beep -> blauw
Alarm 2 -> 4 uur bij de wasmachine -> elke donderdag alleen thuis -> paars

1,1,1,1,1, is hard to differentiate from 0,0,0,0,0

Think aloud method:
I'm going to turn the alarm on for the chosen thing. When this alarm goes off, can you try to turn it off? While doing so please speak about everything you're thinking about.

Note for me: during the performing of this action, note any unexpected behaviour and ask why the user chose to do this?

- Oh a blue light
- What does that mean?
- Oh I need to go to the kitchen
- Where did I leave the beacon?
- What did I have to do again?
- Ah yes, dishes
- I'll put the device back in my pocket.

Why ritme back in pocket? -> same as with mobile phone -> always things in a pocket

Questions:

<u>How easy or difficult was it to understand which location you had to go to?</u>

- Pretty difficult to understand the connection between led and beacon
- The standard blue colour is confusing because the standard blue colour used for a lot of other devices
- Expected something on the bottom to light up -> which sections would be needed would have lights (at the bottom)
-

<u>What keywords come to mind when thinking about this product?</u>

- Form-fitting
- Useful
- Very present
- Would expect the device to be more present in pants due to its size (easier for men)

A strong feeling of having it with you -> not clear whether its negative or positive

<u>If you could change one thing, what would it be?</u>

- The lights at the bottom
- Very easy to put away (which is bad) -> for example, add feet to maps in flippable (negative design)
- The sound should be louder
- Led's are annoying (pretty bright)
- The button is pretty bad (as it is easy to use (to turn off the alarm))
- A mute button would be useful (with a 3-second timer)
    - In case of failure, it would be useful
    - Easy mode and Hard mode models (no mute button)
- How to charge the thing (USB thing vs custom charger)
    - Longer battery life would be great
    - How to reduce energy consumption
    - A reminder of when to charge

Do you have any psychological associations with the interaction of this product? (for example, hanging up your coat signalling the end of a workday)

- There are of course things making you want to not follow the ritme (e.g. laziness)
- Because you bought it you can force yourself to it
    - The only way to turn it off (the entire thing) would be to put it in the charging dock
        - The charging dock still makes noise (do you really want to stop? In the app)

Introduce interaction prototype

Ideal form factor -> doesn't bother me -> even if upside down -> pretty clear in which orientation the device should be in the pocket -> feels very nice
-> subtle texture along the edges to further give an indication of the correct orientation

Questions:

<u>If you were to place the prototype down, in what orientation would you do that?</u>

- The correct way

<u>Compared to the technical prototype, what do you like more or less about the product?</u>

- Nice that it has a curved edge (less sharp)
- Looks less like an alarm clock -> so wouldn't put it away as easily
- Reminds him of a Nintendo 64 cartridges
- Feels unique -> more than the previous thing
- The bigger sized prototype looks more like an alarm
- Beacon easier to put somewhere (because it's smaller)
- The bigger things have more the interaction of a key (because of size) (same size beacon -> smaller device)

<u>Do you prefer the beacon things with edges or without</u>

- Feel more robust with things ->, therefore, easier to put in pockets

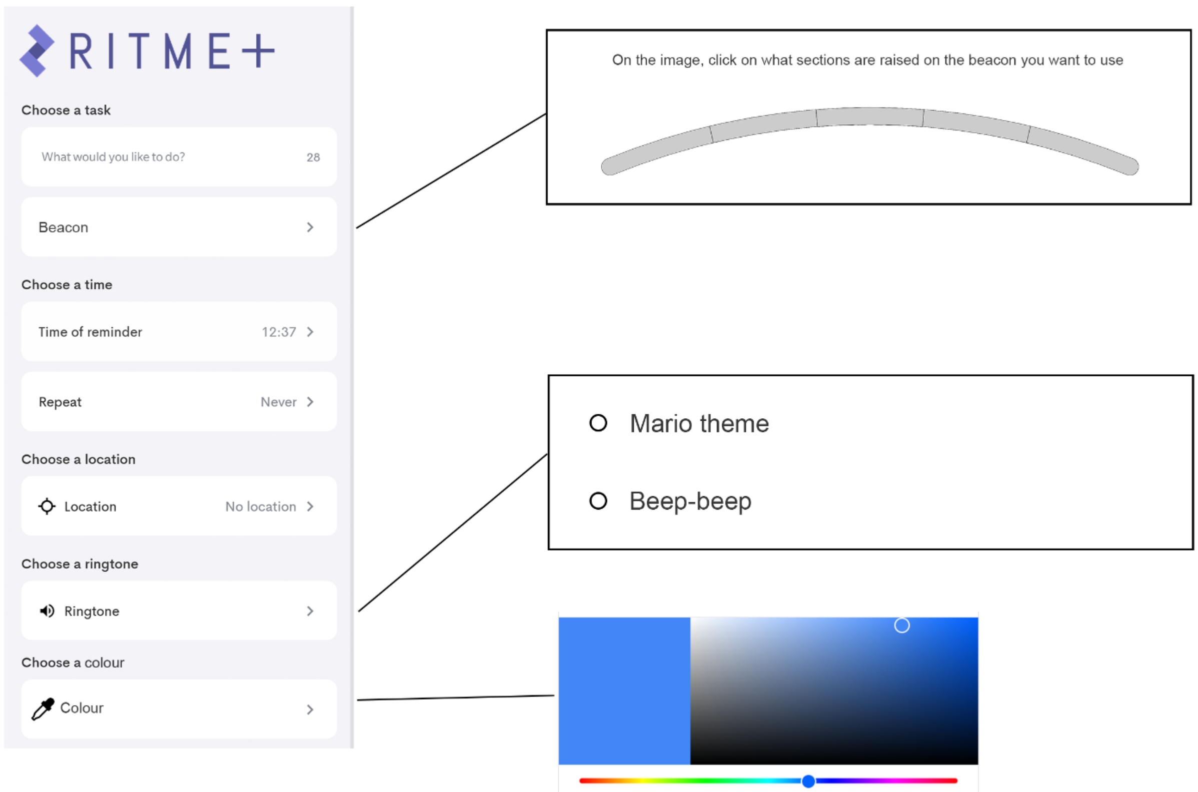
<u>Would you prefer using this product or your phone to turn off the alarm?</u>

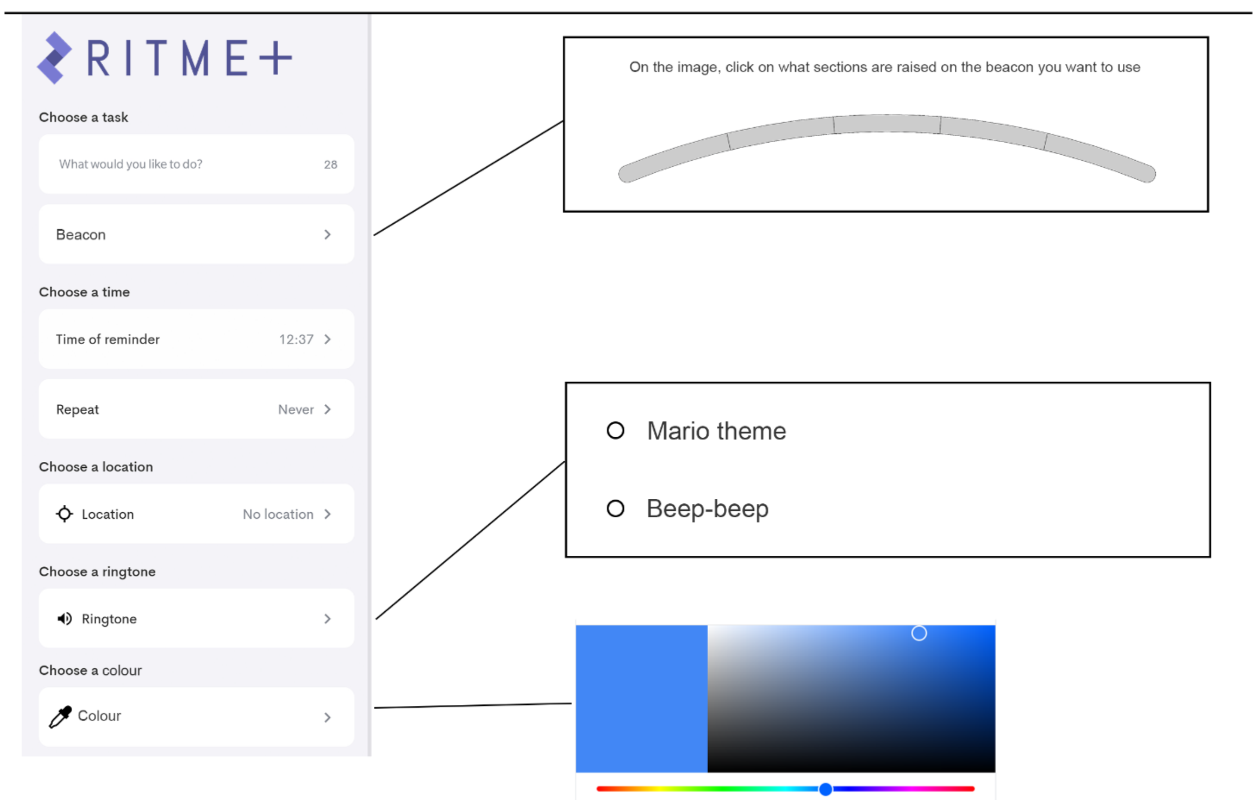
- Not sure yet. Would need long term use -> with an improved version yes. Would have added value compared to QR codes

- The physical thing feels more special
    - Not sure if its because it new
    - After a few weeks, (for example after losing stuff) opinion might change
- Overall pretty positive.

Delegeren van locatie + andere dingen naar device ipv mobiel.

- Most important thing is to do this ^^^^
    - For example, to save energy on the phone, more accuracy, and GPS stops when the screen goes dark.
    - The charging station could possibly check the GPS signal
        - Notify the device that the user is home/ at work by clicking it in the thing (high possibility for error -> should be done autonomously)


RITME+
Choose a task
What would you like to do?
28
Beacon
Choose a time
Time of reminder
12:37
Repeat
Never
Choose a location
Location
No location
Choose a ringtone
Ringtone
Choose a colour
Colour
On the image, click on what sections are raised on the beacon you want to use
Mario theme
Beep-beep


RITME+
Choose a task
What would you like to do?
28
Beacon
Choose a time
Time of reminder
12:37
Repeat
Never
Choose a location
Location
No location
Choose a ringtone
Ringtone
Choose a colour
Colour
On the image, click on what sections are raised on the beacon you want to use
Mario theme
Beep-beep